Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings

LWDA’17

September 11-13, 2017
Rostock, Germany
Preface

LWDA 2017 conference provides a joint forum for experienced and young researchers, to bring insights to recent trends, technologies and applications and to promote interaction in the research field of big data and beyond.

The acronym LWDA expands in German to “Lernen. Wissen. Daten. Analysen.” (English: “Learning. Knowledge. Data. Analytics.”). Recent research in the field is presented and discussed from the viewpoint of machine learning, data mining, knowledge extraction, knowledge management, information retrieval, personalization, database management, information systems, big data management and big data analytics to name a few.

The LWDA conference series comprises the workshops IR, KDML, FGWM and FGDB which are organized by the respective special interest groups within the German Computer Science Society:

- FG-IR 2017 - Information Retrieval
- FG-KDML 2017 - Knowledge Discovery, Data Mining and Machine Learning
- FG-FGWM 2017 - Knowledge Management
- FG-DB 2017 - Database Systems

The papers published in LWDA 2017 proceedings have been selected by independent program committees from the respective fields. The program consists of four invited keynotes and two joint research sessions as well as the community meetings of the special interest groups. In addition to these joint sessions, there are four parallel research sessions for each of the workshops focusing on more specific topics. A joint poster session gives all presenters the opportunity to discuss their work in a broader context. This year’s social program includes a city tour for further interaction on the second evening.

Our distinguished keynote speakers are:

- Prof. Dr. Carsten Binning - TU Darmstadt
- Prof. Dr. Mirjam Minor – University of Frankfurt
- Prof. Dr. Eirini Ntoutsi – University of Hannover
- Welf Wüstlich - Planet AI

The Institute of Business Administration at the University of Rostock is proud to host the LWDA 2017 conference. For the technical program the organizer would like to thank the workshop chairs and their programme committees for their hard work as well as the keynote speakers for their inspiring talks. Finally, we acknowledge the great support of our sponsors futureTV, Data Group and wegtam. We hope the participants will keep the venue as an inspiring event with fruitful discussions in mind and the readers will enjoy studying the scientific contributions in this proceedings volume. The proceedings are published with CEUR and can be found here. http://ceur-ws.org/Vol-1917

Rostock, Germany, September 2017

Michael Leyer
Conference Organization

General Chair

Michael Leyer  University of Rostock

Program Chairs

Christian Bauckhage  Fraunhofer IAIS, St. Augustin
Ulf Brefeld  Leuphana University of Lüneburg
Ingo Frommholz  University of Bedfordshire
Claus-Peter Klas  GESIS - Leibniz Institute for the Social Sciences
Meike Klettke  University of Rostock
Andrea Kohlhase  University of Applied Sciences Neu-Ulm
Eric Kübler  Goethe University Frankfurt
Klaus Meyer-Wegener  Friedrich-Alexander University Erlangen-Nürnberg

Program Committee

Klaus-Dieter Althoff  DFKI / University of Hildesheim
Wolf-Tilo Balke  TU Braunschweig University
Christian Bauckhage  Fraunhofer IAIS
Joachim Baumeister  denkbares GmbH
Dennis Becker  Leuphana University of Lüneburg
Martin Becker  University of Würzburg
Axel Benjamins  Otto (GmbH & Co KG)
Ralph Bergmann  University of Trier
Alexander Böhm  SAP
Ulf Brefeld  Leuphana University of Lüneburg
Vincent Bremer  Leuphana University of Lüneburg
Eduardo Brito  Fraunhofer IAIS
Stefan Conrad  University of Düsseldorf
Stephan Deßloch  TU Kaiserslautern University
Michael Fellmann  University of Rostock
Reginald Ferber  Hochschule Darmstadt
Norbert Fuhr  University of Duisburg-Essen
Marwan Hassani  Eindhoven University of Technology
Andreas Henrich  University of Bamberg
Alexander Hinneburg  Martin-Luther-University Halle-Wittenberg
Andreas Hotho  University of Würzburg
Dimitris Karagiannis  University of Vienna
Kristian Kersting  TU Darmstadt University
Meike Klettke  University of Rostock
Andrea Kohlhase  University of Applied Sciences Neu-Ulm
Ralf Kreßel  Hasso Plattner Institute
Eric Kübler  Goethe University Frankfurt am Main
Florian Lemmerich,  GESIS - Leibniz Institute for the Social Sciences
Ulf Leser  Humboldt-University of Berlin
Edith Maier  University of Applied Sciences St. Gallen
Sebastian Mair  Leuphana University of Lüneburg
Alke Martens   University of Rostock
Christoph Martin   Leuphana University of Lüneburg
Philipp Mayr   GESIS – Leibniz Institute for the Social Sciences
Klaus Meyer-Wegener   Friedrich-Alexander University Erlangen-Nürnberg
Mirjam Minor   Goethe University Frankfurt
Emmanuel Müller   Hasso Plattner Institute
Gilbert Mueller   University of Trier
Peter Mutschke   GESIS – Leibniz Institute for the Social Sciences
Felix Naumann   HPI Potsdam
Nico Piatkowski   TU Dortmund University
Ulrich Reimer   University of Applied Sciences St. Gallen
Pascal Reuss   Universität Hildesheim
Kurt Sandkuhl   University of Rostock
Christian Severin Sauer   University of West London
Petra Sauer   Beuth Hochschule für Technik, Berlin
Philipp Schaer   TH (Köln) University of Applied Sciences
Ralf Schenkel   University of Trier
Ute Schmid   University of Bamberg
Holger Schwarz   University of Stuttgart
Rafet Sifa   Fraunhofer IAIS
Maryam Tavakol   Leuphana University of Lüneburg
Jens Teubner   TU Dortmund University
Matthias Uflacker   HPI Potsdam
Christian Wolff   University of Regensburg
Christa Womser-Hacker   University of Hildesheim
# Table of Contents

Analyzing SQL Query Logs using Multi-Relational Graphs ................................................................. 1
   Andreas M. Wahl and Richard Lenz

Shapley Curves: A new concept for modelling feature importance ..................................................... 2
   Farjad Adnan, Karlson Pfannschmidt and Eyke Hüllermeier

Improving Taxonomy Maintenance: Automated Splitting and Merging of Taxonomies .................... 3
   Moritz Flöter, Pascal Reuss, Johannes Ude and Klaus-Dieter Althoff

Pivot Selection for Dimension Reduction Using Annealing by Increasing Resampling .................... 15
   Yasunobu Imamura, Naoya Higuchi, Tetsuji Kuboyama, Koichi Hirata and Takeshi Shinohara

Towards Learning Structural Node Embeddings using Personalized PageRank ............................... 24
   Felix Borutta, Julian Busch, Evgeniy Faerman and Matthias Schubert

Sentiment analysis of a German Twitter-Corpus ................................................................................. 25
   Malte Flender and Carsten Gips

MixedTrails: Bayesian Hypothesis Comparison Heterogeneous Sequential Data ............................... 37
   Martin Becker, Philipp Singer, Florian Lemmerich, Markus Strohmaier and Andreas Hotho

Predicting Tags for Stack Overflow Questions .................................................................................... 38
   Sayan Pal, Ammar Shaker and Eyke Hüllermeier

Finding Hierarchy of Topics from Twitter Data .................................................................................... 39
   Nghia Duong-Trung, Nicolas Schilling and Lars Schmidt-Thieme

Using Clustering for Categorization of Support Tickets ....................................................................... 51
   Daniel Beneker and Carsten Gips

Robust Spectral Clustering for Noisy Data ............................................................................................ 63
   Aleksandar Bojchevski, Yves Matkovic and Stephan Günnemann

A Human Like Incremental Decision Tree Algorithm: Combining Rule Learning, Pattern Induction, and Storing Examples ................................................................. 64
   Christina Zeller and Ute Schmid

Interpretable Matrix Factorization with Stochastic Nonnegative DEDICOM .............................. 74
   Rafet Sifa, Cesar Ofeda, Kostadin Cvejoski and Christian Bauckhage

PRIMPing Boolean Matrix Factorization by Proximal Alternating Linearized Minimization ............. 86
   Sibylle Hess, Nico Piatkowski and Katharina Morik
KPCA Embeddings: An Unsupervised Approach to Learn Vector Representations of Finite Domain Sequences

Eduardo Brito, Rafet Sifa and Christian Bauckhage

jPL: A Java-based Software Framework for Preference Learning

Pritha Gupta, Alexander Hetzer, Tanja Torende, Sebastian Gottschalk, Andreas Kornelsen, Sebastian Osterbrink, Karlson Pfannschmidt and Eyke Hüllermeier

A Bayesian approach for comparing hypotheses about sequential data and its applications

Florian Lemmerich, Philipp Singer, Martin Becker, Lisette Espín-Noboa, Dimitar Dimitrov, Denis Helic, Andreas Holtho and Markus Strohmaier

A Latent-Feature Plackett-Luce Model for Dyad Ranking Completion

Dirk Schäfer

Learning to Rank based on Analogical Reasoning

Mohsen Ahmadi Fahandar and Eyke Hüllermeier

Semantic Query Processing: Estimating Relational Purity

Jan-Christoph Kalo, Christoph Lofi, René Pascal Maseli and Wolf-Tilo Balke

Leveraging Patient Similarity Analytics in Personalized Medical Decision Support System

Araek Tashkandi and Lena Wiese

De-Anonymisierungsverfahren: Kategorisierung und Anwendung für Datenbankanfragen

Johannes Goltz, Hannes Grunert and Andreas Heuer

Einsatzmöglichkeiten von Crowdsourcing im Umfeld des Mecklenburgischen Flurnamenarchivs

Martin Lichtwark, Holger Meyer and Andreas Heuer

Reconstructing Graph Pattern Matches Using SPARQL

Stephan Mennicke, Denis Nagel, Jan-Christoph Kalo, Niklas Aumann and Wolf-Tilo Balke

Konzepte für das Forschungsdatenmanagement an der Universität Rostock

Ilvio Bruder, Andreas Heuer, Sebastian Schick and Sascha Spors

Bringing computation to the data(base) with Core Data Services (CDS)

Stefan Bäuerle and Alexander Böhm

Enterprise-Datenbank – vergessen, oder?

Jürgen Bittner

Adaptive Business Process Visualization for a Data and Constraint-Based Workflow Approach

Eric Rietzke, Ralph Bergmann and Norbert Kuhn

Irreführende mentale Modelle beim Smart-TV

Gamze Ucar and Andrea Kohlhase
Analyzing SQL Query Logs using Multi-Relational Graphs

Andreas M. Wahl and Richard Lenz

Computer Science 6 (Data Management), FAU Erlangen-Nürnberg

{andreas.wahl|richard.lenz}@fau.de

Analytical SQL queries are a valuable source of information. They contain expert knowledge that cannot be inferred from schemas or content alone. Consider, for example, data lake scenarios, where relational and semi-structured data sources are combined in a single storage and processing environment. Data lakes often lack a structured curation process [1]. Neither global schemas nor vocabularies might be established and data sources might be disparate. SQL provides effective mechanisms to apply these curation steps during querying in a demand-driven way (e.g. by using aliases, joins, casts, user-defined functions, conditional expressions). Hence, the resulting SQL query logs constitute a dynamic documentation of the data lake and the knowledge gathered by its users through previous pay-as-you-go integration tasks. This knowledge includes the purpose of data sources, their semantics, vocabularies, associations with other data sources, and their temporal and social usage context.

To leverage this knowledge, we have developed an extensible framework for analyzing SQL query logs. Query logs are mapped to a multi-relational [3] graph model. We store query texts and corresponding abstract syntax trees to enable meta-querying for syntactic features. However, as SQL allows expressing queries with many different language constructs and the use of aliases, wildcards and unqualified attributes, meta-querying for semantic features requires a different query representation. We convert each query to a corresponding relational algebra tree and normalize it using algebraic transformation rules. Each tree is interlinked with a schema lineage tree, which captures attribute lineage and output schemas of each relational operator. Metadata about users, physical time and logical order allows to inspect the social and temporal context of each query. Meta-queries are specified using domain-specific graph traversal expressions.

Our framework can be used for a broad range of application scenarios. It facilitates collaborative data science by locating relevant queries. Other use cases include maintenance and monitoring tasks, schema evolution mechanisms and existing log mining algorithms. We rely on Apache TinkerPop [2] to abstract from vendor-specific graph implementations. TinkerPop enables both interactive meta-querying and complex distributed computations on our graph model.

References
Shapley Curves: A New Concept for Modelling Feature Importance

Farjad Adnan, Karlson Pfannschmidt, Eyke Hüllermeier
Intelligent Systems Group
Paderborn University

We propose a novel method for measuring the importance and usefulness of predictor variables (features) in supervised machine learning, which makes use of concepts from cooperative game theory. The basic idea of our approach is to consider subsets of variables as coalitions, and their predictive performance as a payoff. This approach acknowledges the fact that the usefulness of a feature in a learning context strongly depends, not only on the learning method being used, but also on the other features being available.

A theoretically appealing measure of the importance of an individual feature is the Shapley value [3]. Computationally, however, this measure is challenging. First, the exact computation of the Shapley values requires determining the performance of all possible subsets of features, which is in general #P-hard [2]. Furthermore, in the context of machine learning, even the training of a single predictor on one subset of features can take a considerable amount of time.

As another aspect specific to machine learning, let us note that the Shapley values of each feature can change with varying sample size, due to effects such as overfitting. Motivated by this observation, we introduce the concept of a Shapley curve, which depicts the (weighted average) contribution of a feature to the learning curve (expected performance as a function of the sample size).

We develop an approximation technique for estimating Shapley values, which is efficient in the number of models that need to be trained and validated. Moreover, to estimate Shapley curves, we propose a hierarchical Bayes approach that does not require an evaluation of all possible subsets of features on different sample sizes. Last but not least, leveraging related techniques for extrapolating learning curves [1], we are able to estimate the Shapley values in the limit when the sample size goes to infinity. We evaluate our approach on synthetic and real-world datasets.

References

Improving Taxonomy Maintenance: Automated Splitting and Merging of Taxonomies

Moritz Flöter\textsuperscript{2}, Pascal Reuss\textsuperscript{1,2}, Johannes Ude\textsuperscript{2}, and Klaus-Dieter Althoff\textsuperscript{1,2}

\textsuperscript{1} German Research Center for Artificial Intelligence
  Kaiserslautern, Germany
  \url{http://www.dfki.de}

\textsuperscript{2} Institute of Computer Science, Intelligent Information Systems Lab
  University of Hildesheim, Hildesheim, Germany
  \url{http://www.uni-hildesheim.de}

Abstract. In Case-Based Reasoning (CBR), taxonomies are often used to model similarities. For complex domains and tasks such taxonomies tend to increase in size making them hard to model and maintain. This especially holds true if a group of people is working on the same taxonomy simultaneously. In this paper, we propose a solution by dividing larger taxonomies into sub-taxonomies, which can be regarded as individual and independent taxonomies. Through compilation, they can be merged and put back into their original context.

1 Introduction

For structured CBR systems, taxonomies are a relatively efficient way of defining similarities between different values of a certain attribute. This is achieved by ordering the possible values in hierarchical form as nodes of a tree and assigning similarity values to each node \cite{2,3}. Beginning from the most general descriptor for the attribute in the root node, the values on each child node get more and more specific towards the leaf nodes of the tree. The similarity values follow that pattern usually starting at 0.0 for the root node and ending at 1.0 for the leaves. The value of each node serves as its unique identifier as the calculation of similarities has to be consistent. Once the tree is complete, the similarity between two different known values of the same attribute is calculated by choosing the closest common predecessor in the tree-structure. In case the attribute-values are identical or if the query contained a more general attribute-value that is a direct predecessor to the value in a case from the case-base, the resulting similarity is 1.0. It should be noted that the same concepts for similarity-calculation remains applicable in our approach for a compiled taxonomy.

Modeling taxonomies can be difficult and can cause a high effort, especially when modeling hundreds of values in a single taxonomy. For example, a taxonomy of aircraft systems may contain several thousand values from components (e.g. cabin) to systems (e.g. in-flight entertainment system), to system parts (e.g. audio system) to so-called Line-Replaceable-Units (LRU, e.g. display, speaker). Modeling each system or system part in an individual taxonomy would reduce
the modeling and maintenance effort, but similarity computation between values of different taxonomies will become very difficult. Therefore, splitting a big taxonomy into smaller sub-taxonomies solves just a part of the effort problem. To compute the similarities between values of different sub-taxonomies, the relevant sub-taxonomies should be merged during retrieval. This way, the modeling and maintenance effort is reduced, while still allowing a comparison of values from different sub-taxonomies.

The motivation for this work has its seeds in the OMAHA research project\cite{4}. In this project we modeled taxonomies for aircraft systems with several thousand nodes. A taxonomy with three subsystems has 7668 nodes. Maintenance of this taxonomies is only possible with high effort. Therefore, the idea is to split up the big taxonomies into several smaller taxonomies with only dozens of nodes. This way we could maintain the smaller taxonomies with less effort. To have the same effect on similarity as the origin taxonomy, the smaller taxonomies have to be merged for similarity measurement.

For the rest of this paper, we will regard taxonomies from their structural point of view, namely trees. When splitting up this structure, there are some obstacles that need to be overcome. It is important that the similarity measure of the sub-taxonomies can be used even when the respective attribute values are put back into their original context. Furthermore, because we are splitting up a structure, that could mean that some parts of the structure are reused at more than one single point. This offers the chance of eliminating redundant work on repetitive parts of the taxonomy, but also forces us to consider means to keep the nodes uniquely identifiable.

This paper is structured as follows, in Section 2 we describe some related work and distinguish our approach from previous research. Section 3 describes in more detail the approach of splitting and merging taxonomies and its implementation within the tool myCBR\cite{1}. We then give an argumentative evaluation and conclude with an outlook on future work.

2 Related Work

There is plenty of research on taxonomies and on combining and merging them, that is not directly related to CBR. Several algorithms have been described to merge different taxonomies and to unify them, such as RCC-5\cite{10} or the Extended Integrated Taxonomy Generation algorithm\cite{8}. These algorithms merge taxonomies from different sources into a target taxonomy. During the merge process duplicate values are removed, sub-taxonomies and leaves are combined, and sometimes unused nodes are removed. The intention of these algorithms is to unify the values of different taxonomies to put them into a common context. Our approach tends not to unify the values of smaller taxonomies, but allows the integration of several taxonomies via reference nodes into another taxonomy. This way we also put the taxonomies into a common context, but with the focus on similarity computation during retrieval.
There is also some research in the context of dynamic similarity measures. Approaches range from dynamic similarity on databases\cite{6} via their use in data mining and clustering\cite{9} to adaptive similarity measures in CBR\cite{7}. These approaches use similarity measures to retrieve data from data sources while dynamically adapting these measures depending on the content of the query. Our approach also allows a dynamic similarity measure, using the dynamic composition of taxonomies for retrieval. This can be done by the user before the retrieval or depending on the content of the query during the retrieval. This allows us to compose a taxonomy as similarity measure for attributes depending on the required information from sub-taxonomies.

3 Splitting and Merging of Taxonomies

In this section we describe our approach for splitting and merging taxonomies and the current implementation in the open source tool myCBR.

3.1 General introduction to the concept of splitting taxonomies

Before splitting up taxonomies and being able to combine them dynamically, it is necessary to model an interface between taxonomies that defines their relation. By assigning a unique identifier to each sub-taxonomy, it is possible to reference them when needed. A node in one taxonomy could for example reference an entire separate taxonomy by its name. As trees contain one single root element, the referencing node therefore gets replaced with the root-node of the referenced taxonomy. If a referenced sub-taxonomy can not be found, for example because it was deleted, the referencing node will be treated as a leaf node.

Because the important part in the context of CBR is the use of taxonomies as similarity measures, the similarity values of the sub-taxonomies have to remain consistent. Maintaining the original values of the original single taxonomy would be the most obvious solution but does defeat the purpose of splitting the taxonomy in the first place. If all similarity-values of the nodes are simply kept, the sub-taxonomies continue to be dependent on the context from which they are referenced from. It should be possible to model the similarity values of each sub-taxonomy individually without that kind of dependency to use a sub-taxonomy as an individual similarity measure. This also allows the same sub-taxonomy to be reused in multiple contexts by multiple different taxonomies - even across different projects - that reference it.

Considering that a typical taxonomy has the similarity value of 0.0 in its root node and that similarity-values have to increase from the root towards the leaves of the taxonomy, a solution to this problem has to adjust the similarity values of the referenced sub-taxonomy when inserting it back into the project-context. When inserting a sub-taxonomy into another taxonomy, the important parameter to define the context is the similarity-value in the referencing node. Therefore, it is also important that a referencing node keeps the value that was originally in place at its position when a taxonomy is split.
3.2 Merging of taxonomies

Once taxonomies are separated it should be possible to merge them together into a single taxonomy that can directly be used in the context of CBR. For the merging procedure we propose the compilation of a combined taxonomy, based on a formula that adjusts the similarity-values of a sub-taxonomy when inserted into a taxonomy that references it.

In the following formula the similarity value of the referencing node is represented by $\text{sim}_{\text{referencing}}$ while $\text{sim}_{\text{referenced,root}}$ represents the value stored for the root node of the referenced taxonomy. Given the old similarity value of any node of the referenced sub-taxonomy $\text{sim}_{\text{old}}$, it is possible to calculate a new similarity-value $\text{sim}_{\text{calculated}}$ for the node in the project-context.

$$\text{sim}_{\text{calculated}} = \text{sim}_{\text{referencing}} + \frac{\text{sim}_{\text{old}} - \text{sim}_{\text{referenced,root}}}{1 - \text{sim}_{\text{referenced,root}}} (1 - \text{sim}_{\text{referencing}})$$

This formula proportionally maps the similarity values of the referenced sub-taxonomy from the interval $[\text{sim}_{\text{referenced,root}}, 1]$ to the interval $[\text{sim}_{\text{referencing}}, 1]$. If $\text{sim}_{\text{referencing}}$ or $\text{sim}_{\text{referenced,root}}$ is already 1.0, all nodes of the sub-taxonomy are inserted into the project-context with the similarity-value of 1.0. The mapping is achieved following the pattern of the mathematical mapping from a source-interval to a target-interval as shown later in this section with $x_{\text{mapped}}$.

Depending on the conventions of a project, one might assume that $\text{sim}_{\text{referenced,root}}$ always carries the value 0.0. This does however not have to be the case. Some projects might choose to keep their existing values when they split taxonomies because the people working on the different parts of the taxonomy are already
used to seeing specific values for their part of the taxonomy. Using the proposed version of the formula, that remains possible while assuming \( \text{sim}_{\text{referenced.root}} = 0.0 \) would offer the possibility to further simplify it.

**Fig. 2.** Calculation of Similarity-Values

![Calculation of Similarity-Values](image)

Given that all nodes below the referencing node have higher or equal calculated similarity-values compared to it, the condition to have increasing similarity-values from top to bottom is fulfilled.

In the following, we describe the merge process of taxonomies. We call the referencing taxonomy \( \text{TaxA} \) while the referenced taxonomy is called \( \text{TaxB} \). The similarity-values of the nodes in \( \text{TaxA} \) remain the same as before. \( \text{TaxB} \) initially starts with a root-node that carries the same similarity-value as the referencing node. All other nodes are assigned the same similarity-values as before.

Because the goal is to be able to look at the taxonomies as separate and independent constructs, the values of \( \text{TaxB} \) are now mapped proportionally from the original interval \([\text{sim}_{\text{referencing}}, 1]\) to \([0, 1]\). \( \text{TaxB} \) now carries similarity-values that are detached from the original project-context.

Finally and most importantly, the resulting values calculated for the project-context are actually **identical** to the values of the original taxonomy. In fact it does not matter to which interval \([b, 1]\) we mapped the original values to as long as \( b \neq 1 \).
Proof (Calculated similarity-values are identical to original after merge). Assuming the original values of a sub-taxonomy were mapped from \([a,1]\) to \([b,1]\), \(a, b \neq 1\) and a given mapped similarity \(\text{sim}_{\text{mapped}}\) of an arbitrary element from that sub-taxonomy, we have to prove for that the calculated similarity based on \(\text{sim}_{\text{mapped}}\) is the same as the original similarity \(\text{sim}_{\text{original}}\) (\(\text{sim}_{\text{calc}} = \text{sim}_{\text{original}}\)).

Note: \(b=a\) is not excluded.

Inserting into the formula, we get

\[
\text{sim}_{\text{calculated}} = a + \frac{\text{sim}_{\text{mapped}} - b}{1 - b} (1 - a)
\]

As proportional mapping from an interval \([g, h]\) to another interval \([i, j]\) follows the scheme

\[
x_{\text{mapped}} = i + \frac{x - g}{h - g} (j - i)
\]

and we originally mapped from \([a,1]\) to \([b,1]\), we get

\[
\text{sim}_{\text{mapped}} = b + \frac{\text{sim}_{\text{original}} - a}{1 - a} (1 - b)
\]

\[
\Rightarrow \text{sim}_{\text{calculated}} = a + \frac{b + \frac{\text{sim}_{\text{original}} - a}{1 - a} (1 - b) - b}{1 - b} (1 - a)
\]

\[
= a + \frac{\text{sim}_{\text{original}} - a}{1 - b} (1 - a) = \text{sim}_{\text{original}}
\]

\[
\square
\]

3.3 Compilation of taxonomies

When implementing the compilation algorithm for sub-taxonomies, the basic structure follows the patterns used for conventional trees. Each node contains a descriptive id which matches an attribute value of the CBR application and a similarity value. A non-leaf node may have children nodes attached directly to it or via a reference to another taxonomy.

In our implementation the decision between children nodes and a reference is exclusive in order to keep the overall structure as simple to understand as possible. This, however, is a pure design decision and does not impose a technical restriction. The decision was made in order to achieve a clear distinction between the two different ways in which children may be included. Nodes that reference another taxonomy are called referencing nodes while others are called non-referencing nodes.

The compiler gets passed a set of sub-taxonomies as resources for the compilation as well as the information on which sub-taxonomy defines the root of
the desired output. Because all of the sub-taxonomies are referenced by their id/name, it makes sense to use a key-value mapping for efficient access to the requested sub-taxonomy.

When inserting the same taxonomy multiple times into one big taxonomy at different referencing nodes, the identification of the different instances of the sub-taxonomy needs to remain unambiguous. For those use-cases, the compiler offers the option to generate unique names for each of the nodes by inserting the name of the referencing nodes parent as a prefix before the names of the nodes in the referenced taxonomy. Using ':' as a separator symbol between the names, the names for the taxonomy-nodes from 'other mobile devices' as shown in Figure 2 would result in 'mobile:other mobile devices', 'mobile:Smartphone' and 'mobile:Tablet'.

Figure 3 shows an overview of the algorithm executed in order to compile the taxonomy. It should be noted that the algorithm treats directly attached children and children that are inserted by reference in exactly the same way. A referencing node gets replaced by the node that it references and the children of that referenced node are considered its children.
3.4 Runtime depending on input size

We define the number of nodes of the resulting taxonomy as input length $n$. This definition of the input length is somewhat unconventional as it assumes information about the resulting taxonomy. It still makes sense though when put into the context of our concept. The alternative to this algorithm and concept is to have large singular taxonomies with all their disadvantages. The concept of subtaxonomies offers the possibility to represent the same taxonomies in a simplified way - therefore taking the resulting singular taxonomy as input size is not as far fetched as it might seem at first. The basic processing of each node happens in $O(1)$ as the only action performed for each node is the adjustment of its values. Each node needs to be processed one time resulting in $O(n)$ for the average, worst and best case. The part where a non-linear increase of the runtime is possible would be the process of finding the according taxonomy for a reference. As mentioned before, an access-optimize storage model for access from a given name/identificator of a taxonomy would make sense here. The expected effort for finding elements by using hashing is within $O(1)$ depending on the amount of elements in the hashtable compared to the size of the hashtable [5, Chapter 6.4]. While the worst case can be $O(n)$, optimized hybrid implementations like Java 8’s HashMap offer a runtime of $O(\log(n))$ for the worst case by using search trees.

The algorithm is able to efficiently compile a taxonomy within an expected runtime of $O(n)$. In the end even the worst case runtime of the algorithm is in $O(n \times \log(n))$. It should be noted that this would assume that the largest part of nodes are referencing nodes, which does not make sense in real life scenarios for obvious reasons. Assuming a low (compared to the number of nodes in total) and constant number of referencing nodes we further approach $O(n)$ even when hitting the worst case for the hashtable.

3.5 Implementation within myCBR

myCBR is an open source tool and library for CBR modeling and retrieval tasks that integrates structural CBR into an easy to use interface, the myCBR workbench and can also be accessed through an API. It is a joint effort of the Competence Center CBR at DFKI, Germany, and the School of Computing and Technology at UWL, UK. Various industry partners like Airbus and Lufthansa Industry Solutions work together with those institutes to explore new possibilities and approaches to CBR systems[4].

Before the integration of sub-taxonomies into myCBR, the tool already offered support for traditional taxonomies that was used as the basis for the integration. As a matter of fact, the integration was achieved while keeping the existing calculations for similarities between cases intact. Sub-taxonomies get compiled into a construct that remains largely unchanged from the original structure which was already used before for the calculation of similarities between cases.
While adding the functionality for sub-taxonomies along with cycle-detection during the compilation of sub-taxonomies to the SDK, the user interface also had to be adjusted.

We created a separate perspective for sub-taxonomies in the workbench, where all sub-taxonomies are collected. The tool myCBR separates attributes and similarity functions in a way that different types of similarity functions can be assigned to an attribute. The addition of the function-type ‘Sub-taxonomy’ allows to assign a single sub-taxonomy to an attribute that is then compiled to a complete taxonomy using all other sub-taxonomies that exist in the project.

Traditional taxonomies that existed in older myCBR projects can be imported as sub-taxonomies. The export of single sub-taxonomies is also possible to transfer them to other myCBR projects. Those taxonomies can be split according to their logical parts (e.g. every system inside of a plane gets its own sub-taxonomy for the plane-taxonomy). However, it serves the purpose of lowering the effort for maintenance to pay special attention to duplicate (sub)parts inside of existing taxonomies.

Processing a traditional taxonomy to gain a set of sub-taxonomies that eliminates duplicate information is a workflow which consists of three steps. First we check if at least one of the imported taxonomies can be simplified. A taxonomy can be simplified if there is repetitive information across several taxonomies or within the same taxonomy. If there are several taxonomies that can be simplified, we use the new myCBR functions to split the repetitive part from the original taxonomy. Then we use set reference to replace the according part in all other taxonomies. After this process we check again if the taxonomies can be simplified even further. If no further simplification is possible, the workflow is finished. Figure 4 shows an event-driven process chain for this workflow.

**Fig. 4.** Workflow to simplify taxonomies with myCBR Workbench

![Workflow Image](image)

Figure 5 shows the interface of the myCBR Workbench in the ‘Sub-taxonomies’ perspective. In order to use a sub-taxonomy for the calculation of similarities for an attribute, the user has to switch to the ‘Modeling’ perspective and add a Sub-taxonomy-Function as a similarity function to an attribute inside of the

---

3 For example: TaxA references TaxB and TaxB references TaxA resulting in a circle
current project. By assigning a Sub-taxonomy-Function as similarity function, the user can now choose any of the taxonomies contained in the collection of sub-taxonomies inside of the project. That taxonomy will be used as the root taxonomy for the compilation of a single taxonomy out of all referenced sub-taxonomies.

As a result of the selection, the taxonomy gets compiled and myCBR can now perform the same similarity-calculations on that taxonomy that were possible before, using traditional taxonomies. In contrast to traditional taxonomies, this compiled version of sub-taxonomies can not be modified directly. All modifications have to be performed directly on the sub-taxonomies which it consists of. After modification of one or multiple of those sub-taxonomies, the user can choose to recompile this taxonomy at any time by clicking ”Reload from Sub-taxonomies”.

4 Evaluation

Following the principle of separation of concerns it makes sense to split up a taxonomy into multiple meaningful parts. Being able to regard each resulting part separately allows for individual modeling thereby making work easier for the ones working on the different parts. Without using the concept of sub-taxonomies
proposed in this paper, a change in a parent node needs to be reflected in all the nodes that succeed it. In our concept, each sub-taxonomy that is integrated under the according node through compilation does not need to be touched. In instances where parts of a taxonomy are reused across multiple projects or multiple taxonomies in the same project, there is also less modeling effort as the changes do not need to be copied over through multiple taxonomies anymore and can instead be implemented by one single sub-taxonomy.

While we already demonstrated the feasibility of using sub-taxonomies through mathematical means earlier in this paper, we also computed multiple random splits of taxonomies provided by airbus at 20 % of each taxonomies nodes, mapped each resulting sub-taxonomy from \([sim_{root}, 1]\) to \([0, 1]\) and finally compared the resulting compiled taxonomies to the original taxonomy checking for deviations.

This process was repeated 10 000 times for the following taxonomies of the OMAHA research project resulting in the processing of 1 020 000 nodes in all iterations combined:

- **tax_engine_type** was split 6 times (Taxonomy size:30 nodes)
- **sim_tax_ata** was split 10 times (Taxonomy size:52 nodes)
- **sim_tax_emitter** was split 3 times (Taxonomy size:18 nodes)
- **default function** was split 0 times (Taxonomy size:2 nodes)

Alongside consistency checks on the general structure of the taxonomy (naming, tree structure), we also measured the average and maximum error between the nodes of the compiled taxonomy compared to its original counterpart. As expected, the results showed no measurable deviation from the original taxonomy.

### 5 Summary and outlook

In this paper, we presented a concept that offers the functionality to combine separately created and maintained sub-taxonomies into a single taxonomy. The feasibility of this concept has been mathematically proven and demonstrated

\[\text{Mapping} \rightarrow \text{Normalization of the sub-taxonomy}\]
with the integration into the myCBR-project. It was further shown that the concept does not impose any noteworthy computational overhead.

During the design we aimed for a concept of taxonomies that allows for efficient maintenance and we assume that the adaption of this concept would bring benefits to any area where complex and hard to maintain taxonomies are prevalent. It is possible to separate complex taxonomies with relatively low effort and risk as similarity values can be kept identical before and after this process. On top of that our concept makes sharing parts of a taxonomy across one or multiple projects possible and comfortable.

For the future an evaluation of this concept with respect to the maintenance effort after and before introduction together with industry partners would be desirable.

References

Pivot Selection for Dimension Reduction Using Annealing by Increasing Resampling

Yasunobu Imamura¹, Naoya Higuchi¹, Tetsuji Kuboyama², Kouichi Hirata¹ and Takeshi Shinohara¹

¹ Kyushu Institute of Technology, Kawazu 680–4, Iizuka 820–8502, Japan
² Gakushuin University, Mejiro 1–5–1, Toshima, Tokyo 171–8588, Japan
imamura.kit@gmail.com

Abstract. In order to select an optimal set of pivots for dimension reduction, such as Simple-Map and sketches based on ball partitioning, we propose a method named Annealing by Increasing Resampling (AIR, for short). AIR assumes that every state is evaluated by using a sample set. Starting from an arbitrary initial state, AIR repeats to transit states by hill climbing, with evaluating the resampled sets whose size initially is small and gradually increases. Experiments verify that AIR can find better sets of pivots than the conventional method and in shorter time than simulated annealing.

Keywords: Similarity Search, Dimension Reduction, Pivot Selection, Simulated Annealing, Annealing by Increasing Resampling.

1 Introduction

Similarity search is one of the most important tasks for information retrieval of multidimensional data. In this paper, we deal with similarity search in metric spaces, where objects within smaller distance are considered similar. Thus, similarity search is a task to find objects near to a given query object.

When the dimensionality of objects is \( m \), the computational cost to measure distance between two objects is \( O(m) \), and when the number of database objects is \( n \), a naïve similarity search by sequential manner needs \( O(nm) \) cost, which is unrealistic for larger \( m \) and \( n \). In order to weaken the effect of \( n \), hierarchical index structures such as R-Tree [1] and M-Tree [2, 3] have been developed. On the other hand, the dimension reduction is a method to avoid influence of \( m \).

Dimension reductions for Euclidean spaces include K-L transformation (or principal component analysis, PCA) [4] and FastMap [5]. On the other hand, dimension reductions such as H-Map [6] and Simple-Map (S-Map) [7] are applicable to any metric spaces metricized by \( L_1 \) distance, Hamming distance, string edit distance and so on [8].

* This work is partially supported by Grant-in-Aid for Scientific Research 17H00762, 16H02870, 16H01743 and 15K12102 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.
Sketches [9–13] are representations of objects in multidimensional data by compact bit strings to reduce search spaces. In conventional search with sketches, Hamming distance between sketches is adopted. However, the mapping to sketches on Hamming distance can never imply a dimension reduction. On the other hand, since quantization of S-Map based on $L_p$ distance [14] is regarded as a kind of sketches and provides a method to compute the distance lower bound between queries and sketches, it can provide the sketch mapping implying a dimension reduction.

The dimension reduction not only reduces distance computation cost but also avoids so-called “the curse of dimensionality”. For example, it is known that the efficiency of R-Tree is decreasing when the dimension is increasing, but the performance can be improved if R-Tree is constructed on projected objects into lower dimensional space by S-Map.

In the dimension reduction, the object is projected to a low dimensional data or a compact bit string so that the projected distance does not extend with respect to the original distance. Although the projected objects of low dimensions cannot completely maintain the original distance relationship, it is important to reduce the information loss. Because the projection distance does not extend the original distance, it is guaranteed that distant objects in the projection space are far from the original space, so “safely pruning” can be done by searching in the projection space. However, if the shrinkage of the distance is large, the object outside the retrieval range actually becomes closer in the projection space, resulting in deterioration in retrieval efficiency. For PCA, analytically optimal projection can be obtained. On the other hand, for H-Map, S-Map and sketches, it has been known no analytically optimal solution, and therefore, it is necessary to use random selection with evaluation function as a clue or heuristic method such as annealing method.

In S-Map, the reference object is selected as a pivot [15–17], and the distance between each object and the pivot is set as a coordinate value, thereby the number of coordinates is given as the number of pivots. Then, the number of pivots at this time is the dimensionality of the projection space and the distance between objects in the projection space is given as the $L_\infty$ distance. Here, a ball partitioning (BP) is to assign 0 and 1 to the inside and the outside of a ball of radius $r$ centered on the reference object $p$, respectively. Then, the sketch using BP can be regarded as the quantization of S-Map image to 0 or 1 depending on whether the distance from the pivot $p$ of the S-Map is not less than the radius $r$ [14].

Note that conventional search methods such as random selection, local search and simulated annealing and binary quantization method using distribution characteristics of data [18] have been adopted to search a set of pivots for S-Map and BP sketches. All of them are optimized by evaluating values concerned with samples. In the S-Map, the distance preservation ratio is adopted as an evaluation value to maximize it. In BP, the collision probability is adopted as the evaluation value to minimize it. In this paper, we propose a new method named annealing by increasing resampling (AIR) as an optimization method and verify the effectiveness in pivot selection.

The simulated annealing (SA) is a search method to transit stochastically according to temperature with evaluating values from the current provisional solution to its neighbor. At the beginning, it starts from a state of high temperature and gradually
lowers the temperature. At high temperature, the probability of transition to low evaluation value is high. When it has low temperature, it transits only according to the evaluation value, that is, it behaves as a local search. On the other hand, this paper proposes a method named Annealing by Increasing Resampling (AIR, for short), where a hill climbing is carried out by using subsample resampled from the sample used for evaluation, and the resampling number is gradually increased. While the number of resampling is small, the evaluation error for the entire sample is large, so the probability of making a transition to a low evaluation is high. That is, the transition using a small number of samples is similar to the random transition at high temperature in SA. As the number of resampling increases, the error of evaluation gradually decreases and approaches the local search. In this way, the behavior of AIR is very similar to SA.

Empirically, in order to obtain a good solution in a wide area by SA, it is necessary to increase the number of transitions at high temperature, so it takes much time to process at high temperature. On the other hand, in AIR, process to high temperature in SA corresponds to transition using a small number of resamples, and evaluation with a small number of samples is low in cost. Therefore, AIR is possible to realize processing at high temperature in low cost, which needs high cost in conventional SA.

2 Preliminaries

In this section, we briefly introduce dimension reduction, Simple-Map, and ball partitioning (BP) sketch to which the optimization method proposed in this paper is applied.

Let \((U, D)\) and \((U', D')\) be two metric spaces, where \(D\) and \(D'\) are distance functions satisfying the triangle inequality. The dimensionality of data \(x\) is denoted \(\text{dim}(x)\). A mapping \(\varphi: U \rightarrow U'\) is called a dimension reduction, if the following conditions are satisfied for any \(x, y \in U\).

\[
\text{dim}(\varphi(x)) \leq \text{dim}(x) \quad \text{(1)}
\]

\[
D'(\varphi(x), \varphi(y)) \leq D(x, y) \quad \text{(2)}
\]

Condition (1) means that it reduces the dimensionality, and condition (2) means that \(D'\) gives a lower bound of \(D\), respectively.

A Simple-Map (S-Map) is based on the projection \(\varphi_p\), using a point \(p\) called a pivot, defined as follows.

\[
\varphi_p(x) = D(p, x)
\]

From the triangle inequality, the following formula holds for \(x, y \in U\).

\[
\|\varphi_p(x) - \varphi_p(y)\| \leq D(x, y)
\]

Using a set \(P = \{p_1, \ldots, p_m\}\) of pivots, we define an S-Map \(\varphi_P\) and a distance \(D'\) as follows.

\[
\varphi_P(x) = (\varphi_{p_1}(x), \ldots, \varphi_{p_m}(x))
\]

\[
D'(\varphi_P(x), \varphi_P(y)) = \max_{i=1}^m \|\varphi_{p_i}(x) - \varphi_{p_i}(y)\|
\]
Thus, when $m$ is smaller than the original dimension, $\varphi_p$ becomes a dimension reduction.

Projecting objects with S-Map, the distance between them may shrink. This shrinkage, that is, the distance deficiency, is desired to be small for similarity search. Increasing the projective dimension reduces the shrinkage of the distance, but it is strongly influenced by “the curse of dimensionality.” Thus, it is important to minimize the shrinkage of the distance in a lower dimension. The distance preservation ratio for a set $S$ of pairs $(x_i, y_i)$ of points is the following ratio of sums of distances.

$$\frac{\sum D(\varphi(x_i), \varphi(y_i))}{\sum D(x_i, y_i)}$$

Sketches [9–13] are compact bit sequences representing multidimensional data. In this paper, we consider sketches based on ball partitioning (BP). A pivot for BP is a pair $(p, r)$ of a point $p$ and a radius $r$. A BP projection $\sigma_{(p, r)}$ using a pivot $(p, r)$ is defined as follows.

$$\sigma_{(p, r)}(x) = \begin{cases} 0 & \text{if } D(p, x) \leq r, \\ 1 & \text{otherwise}. \end{cases}$$

A sketch mapping $\sigma_p$ of width $w$ bits is defined by a set of pivots $P = \{(p_1, r_1), ..., (p_w, r_w)\}$ as follows.

$$\sigma_p(x) = \sigma_{(p_1, r_1)}(x) \ldots \sigma_{(p_w, r_w)}(x)$$

For example, let consider 4 points $A$, $B$, $C$ and $D$ in a Euclidian plane as in Figure 1. Then, sketches using pivots $P = \{(p_1, r_1), (p_2, r_2)\}$ are $\sigma_p(A) = 01$, $\sigma_p(B) = 00$, $\sigma_p(C) = 10$ and $\sigma_p(A) = 11$.

The conventional similarity search using sketches consists of two stages. First, candidates are selected based on Hamming distances between sketches. Then, the

![Fig. 1. Sketches using two balls](image)
answer is selected from candidates using actual distance. As long as Hamming distance is used, sketch mapping can never imply a dimension reduction. Ohno et al. [14] proposed a method to compute lower bound of distance using sketches. Therefore, such a sketch mapping can imply a kind of dimension reduction. We use the collision probability of a sketch mapping using a set $P$ of pivots as the evaluation function in optimization. We say a collision occurs when two distinct points share the same sketch.

3 Annealing by Increasing Resampling

First we give several assumptions for optimization problem. Let $\Phi$ be the search space of possible solutions. We call an element of $\Phi$ a state. A cost function $f$ gives the evaluation value of a state $x$ with respect to a sample set $S$. Roughly speaking, an optimization problem is to find a solution $x$ from $\Phi$ whose evaluation value is the smallest. The cost function $f$ is desirable to satisfy the following formula for any sample sets $S_1, S_2 \subseteq S$ and any state $x \in \Phi$.

$$\min \left( f(S_1, x), f(S_2, x) \right) \leq f(S_1 \cup S_2, x) \leq \max \left( f(S_1, x), f(S_2, x) \right)$$

(3)

For example, if $f$ is defined by the average of evaluation values for individual samples such as the distance preservation ratio of S-Map, $f$ satisfies the formula (3). The collision probability of sketch approximately satisfies the formula (3) except smaller sample sets. Further, we assume that the neighbor $N(x)$ of a state $x$ always satisfies the following statement.

$$\forall x, y \in \Phi, y \in N^*(x)$$

Here, $N^*$ is a reflexive and transitive closure of $N$. Then, the above statement claims that we can get any state $y$ from any $x$ by finitely many applications of $N$.

We present the algorithm of Annealing by Increasing Resampling (AIR, for short) in Figure 2. Here, the iteration number $i$ of the loop is considered as time, $t: \mathbb{N} \to (0, 1]$ is a monotonic increasing function to give the ratio of resampling number with respect to total samples $S$ and $T$ is the total number of state transitions. Note that resampling

```
function AIR(S: samples): state;
begin
    x := any state;
    for $i := 1$ to $T$
        begin
            $R :=$ randomly selected samples of size $t(i) \times |S|$ from $S$;
            if $f(x, R) > f(y, R)$ for some $y \in N(x)$ then
                $x := y$;
        end
    return $x$;
end
```

Fig. 2. Algorithm AIR
at the $i$-th iteration should be done independently to the preceding resampling. Because it is not appropriate for AIR to make the larger set by adding samples to the previous smaller set in incremental manner.

Note that, when $t(i) = 1$ for any $i$, AIR always uses total samples for state evaluation, thus, it behaves like so called local search. We do not care about detail of method to select a state from the neighbor $N(x)$. In practice, we may select a state with the best evaluation value within a subset of $N(x)$ in a steepest descent manner.

Since, at the beginning stage, the number of resampled samples $R$ is small, the error of $f(x, R)$ with respect to $f(x, S)$ becomes large with high probability, and therefore, AIR may make state transition to a state with lower evaluation value. Thus, AIR makes random walks as SA at high temperature. Finally when $t(i)$ becomes close to 1, AIR behaves as local search because $f(x, R) \approx f(x, S)$.

As the advantage of AIR, it can make search at the beginning stage faster, because state evaluation using smaller samples can be done in low cost. On the other hand, a conventional SA needs high cost for state transitions in high temperature. There is no significant difference of AIR and SA in convergence speed, because AIR can behave almost same as SA by using resampling sizes corresponding to the annealing schedule.

4 EXPERIMENTS

In this section, we give experimental results on optimization of dimension reductions S-Map and BP sketch by the proposed method. We use two kinds of data, feature data of images (images) and SISAP colors database (colors). The number of data in images is 6.8 million extracted from 1,700 videos and dimensionality $n$ of data in images is 64. On the other hand, the number of data in is about 0.1 million and dimensionality $n$ of data in colors is 112. For both data of images and colors, each axis has integer value from 0 to 255 and distances between them are $L_1$.

4.1 Simple-Map

In this experiment, we adopt $m = 8$ for the dimensionality of S-Map, which shows the best performance in similarity search using R-Tree constructed by S-Map images. We use the average value (Ave.) and the standard derivation (S.D.) for distance preservation ratio (DPR) to evaluate pivot sets using randomly selected 5,000 pairs of features. AIR finds a pivot set with maximum distance preservation ratio. A pivot set $P = \{p_1, \ldots, p_n\}$ consists of $mn$ integers corresponding to $m$ pivots of $n$ dimension. The neighbor $N(P)$ of a pivot set $P$ is defined as the set of pivot sets such that any $P'$ in $N(P)$ is the same as $P$ but at one of $mn$ integers. For data of images, features consist of 8-bit integers from 0 to 255. Therefore, $N(P)$ consists of $256mn = 256 \times 8 \times 64$ pivot sets. In our experiments, we implement AIR to randomly choose one of $mn$ integers of $P$, change it from 0 to 255, and move to the best of 256 neighbors of $P$. That is, AIR makes a hill climbing using subsets of neighbors.
We compare AIR with conventional simulated annealing (SA), binary quantization (BQ)[18] and local search (LS). BQ is a heuristic method using stochastic property of data which can find relatively good pivot set within a small computation time. Table 1 shows the results for images. We repeat each method at 10 times. The computing times of BQ and LS are about 50 and 100 seconds, respectively. For SA and AIR, we tuned parameters of the number of state transition trials, which is corresponding to $T$ in Figure 2, to compare their computing times with BQ and LS. We also run SA and AIR in about 500 seconds.

From Table 1, we can observe that AIR can find better pivot sets than BQ in about 50 seconds and LS in about 100 seconds. On the other hand, pivot sets by SA are almost comparable with BQ and LS. For every case of computing time about 50, 100 and 500 seconds, the number of state transition trials by AIR is about 8 times as large as one by SA. This experimentally shows the AIR’s merit to SA pointed out in Section 3.

From Table 2, which shows the results for colors, we can observe the similar behavior of AIR to images in Table 1.

### Table 1. Results for Simple-Map (images)

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (sec)</th>
<th>Trials ($\times 10^3$)</th>
<th>DPR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ave.</td>
</tr>
<tr>
<td>BQ</td>
<td>47.9</td>
<td>—</td>
<td>56.5</td>
</tr>
<tr>
<td>LS</td>
<td>95.3</td>
<td>—</td>
<td>56.1</td>
</tr>
<tr>
<td>SA</td>
<td>49.2</td>
<td>3</td>
<td>56.5</td>
</tr>
<tr>
<td></td>
<td>98.0</td>
<td>7</td>
<td>56.9</td>
</tr>
<tr>
<td></td>
<td>511</td>
<td>40</td>
<td>57.4</td>
</tr>
<tr>
<td>AIR</td>
<td>47.5</td>
<td>24</td>
<td>57.3</td>
</tr>
<tr>
<td></td>
<td>94.2</td>
<td>56</td>
<td>57.4</td>
</tr>
<tr>
<td></td>
<td>487</td>
<td>330</td>
<td>57.5</td>
</tr>
</tbody>
</table>

### Table 2. Results for Simple-Map (colors)

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (sec)</th>
<th>Trials ($\times 10^3$)</th>
<th>DPR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ave.</td>
</tr>
<tr>
<td>BQ</td>
<td>84.6</td>
<td>—</td>
<td>83.2</td>
</tr>
<tr>
<td>LS</td>
<td>196</td>
<td>—</td>
<td>83.6</td>
</tr>
<tr>
<td>SA</td>
<td>85.7</td>
<td>3</td>
<td>83.2</td>
</tr>
<tr>
<td></td>
<td>167</td>
<td>7</td>
<td>83.4</td>
</tr>
<tr>
<td></td>
<td>858</td>
<td>40</td>
<td>83.6</td>
</tr>
<tr>
<td>AIR</td>
<td>85.0</td>
<td>24</td>
<td>83.7</td>
</tr>
<tr>
<td></td>
<td>167</td>
<td>56</td>
<td>83.8</td>
</tr>
<tr>
<td></td>
<td>880</td>
<td>330</td>
<td>83.9</td>
</tr>
</tbody>
</table>
4.2 Sketches

In this experiment, we adopt \( w = 32 \) bits as the width of sketch. Neighbors of pivot set are similarly defined as for S-Map. Radius of a pivot is selected to equally divide space by the ball. The set \( S \) of samples for evaluating pivot sets consists of randomly selected 10,000 points from database. We use collision probability (CP) to evaluate pivot set to be minimized.

We compare AIR with a conventional ball partitioning with random selection (BP), BP using binary quantization (QBP). As for observation of the search performance, we show their precision. Nearest neighbor search using sketches consists of two stages. At first stage, we select candidates using Hamming distance, that is, we select the top \( K \) nearest data in the meaning of Hamming distance. At the second stage, we select the nearest neighbor from the \( K \) candidates. Search precision is the probability that top \( K \) candidates include the exact nearest neighbor. For both databases images and colors, we adopt \( K \) as the 0.1\% of the database size, which is reasonable from both viewpoints of speed and precision.

Table 3 and 4 show results for sketches on images and colors, respectively.

5 Concluding Remarks

In this paper, we have proposed a method of Annealing by Increasing Resampling (AIR, for short) to select an optimal set of pivots for dimension reduction. As shown in Table 1, 2, 3 and 4, AIR can efficiently find better sets of pivots than the conventional method from the viewpoint of evaluation function used for optimization. However, from both Table 3 and 4, from the viewpoint of search precision, the best pivot set is found by the conventional method QBP. However, this is completely the matter of evaluation function. It is a future work to explain the behavior of AIR theoretically. For example, we expect that the solution found by AIR will eventually converge to the optimum one. It is also an important future work for similarity search to inves-

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (sec)</th>
<th>CP (×10^{-6})</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>116</td>
<td>2.6</td>
<td>95.2</td>
</tr>
<tr>
<td>QBP</td>
<td>106</td>
<td>2.2</td>
<td>96.6</td>
</tr>
<tr>
<td>AIR</td>
<td>97.8</td>
<td>1.0</td>
<td>94.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (sec)</th>
<th>CP (×10^{-5})</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>174</td>
<td>3.4</td>
<td>74.7</td>
</tr>
<tr>
<td>QBP</td>
<td>163</td>
<td>7.4</td>
<td>86.4</td>
</tr>
<tr>
<td>AIR</td>
<td>306</td>
<td>1.3</td>
<td>67.3</td>
</tr>
</tbody>
</table>
tigate other evaluation functions than distance preservation ratio for S-Map and collision probability for sketch.

References

Towards Learning Structural Node Embeddings using Personalized PageRank

Felix Borutta, Julian Busch, Evgeniy Faerman, Matthias Schubert

Ludwig-Maximilians-Universität München

In this abstract we present our work in progress on embeddings for nodes in graphs representing their structural similarities. Intuitively, the more similar the structural representations of the respective neighbors of two nodes, the more structurally similar they are. An exemplary application is given by the task of identifying the role of each node in the graph, which might for instance correspond to the function of a protein within a protein-protein interaction network. Modeling and developing formal definitions for concepts such as structural similarity, structural identity and roles remain challenging problems. Existing works either learn embeddings which are not able to appropriately model structural similarity or identify roles based on hand-crafted features rather than learning important features directly from the input graph. Recent work\(^1\) proposes a framework for learning a structural embedding vector for each node based on the degree sequences of its neighbors within a \(k\)-hop distance. Two major drawbacks of this approach are the cubic time complexity and the fact that different \(k\)-hop neighborhoods are considered equally important. Note that in most cases the closer neighbors are considered more important. In contrast, we compute Personalized PageRank (PPR) for each node in the graph and use the resulting vectors as node representations. PPRs are very fast to compute and effectively capture the probability distribution over the corresponding neighborhoods. The preliminary results in Table 1 prove our intuition: we simply sorted the raw PPR vector of each node and used them as representations. However, the experiments used in\(^1\) are of small practical relevance, as they use strongly artificial datasets. Therefore, we also work on an appropriate evaluation framework.

<table>
<thead>
<tr>
<th>Network</th>
<th>complete graph</th>
<th>removed 30% of edges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>struc2vec</td>
<td>ppr</td>
</tr>
<tr>
<td></td>
<td>corresp.</td>
<td>all</td>
</tr>
<tr>
<td>Karate</td>
<td>0.012 ± 0.007</td>
<td>1.517 ± 0.025</td>
</tr>
<tr>
<td>Barbell</td>
<td>0.008 ± 0.005</td>
<td>1.717 ± 0.017</td>
</tr>
<tr>
<td>PPI</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1. Experimental results on three mirrored networks with each having one bridging edge; the graph setting and setup for struc2vec is the same as in\(^1\). ‘corresp.’ shows the mean cosine distances ± the standard deviation between the node embeddings of the corresponding nodes, ‘all’ shows the corresponding values when the distance is measured pairwise between all embeddings. Note that struc2vec did not terminate after 3 weeks for the mirrored PPI network (7780 nodes, 77479 edges).

\(^1\) struc2vec: Learning Node Representations from Structural Identity. Figueiredo, Daniel R and Ribeiro, Leonardo FR and Saverese, Pedro HP. accepted for KDD’17
Sentiment analysis of a German Twitter-Corpus

Malte Flender, Carsten Gips

FH Bielefeld - Campus Minden,
Artilleriestraße 9, 32427 Minden, Germany
{malte.flender,carsten.gips}@fh-bielefeld.de
http://www.fh-bielefeld.de/

Abstract. The amount of easily accessible texts in different languages on the internet grows daily and so the effort and need to organize these texts grows as well. An automated process is needed to extract useful information, like a sentiment, from this amount of published text. This paper deals with the extraction of a sentiment, divided into three classes, from tweets written in German language. Different machine learning algorithms and a variety of preprocessing steps are compared to find the optimal combination. While most work in this field aims at English-language tweets, this paper adapts and transfers these ideas to German-language tweets. The results are compared to the findings of other projects designed for English-language tweets. Further it is shown that the impact of the chosen feature encoding on the results is most significant.

Keywords: Twitter, Sentiment Analysis, Machine Learning, Classification, German

1 Introduction

The popularity of online-social-media, especially microblogging services like Twitter, has vastly increased in the last few years. The idea behind microblogging services is, that a person can write a short statement and post it online. These statements within Twitter are called tweets. A tweet can be shared with and seen by the writer's friends and community.

These tweets often contain a sentiment regarding a certain topic. A single tweet on its own is not very expressive. But if a (topic related) web-crawler collects a significant amount of these tweets, a sentiment analysis can provide useful information. For example, the sentiment of potential customers regarding a certain product, like a smart phone, or the sentiment towards a whole company. Depending on the topic, this information can be used in a wide variety of applications.

In contrast to other forms of text, e.g. movie reviews [7], a tweet is extremely short, at most 140 characters; the language used is so informal that it often contains spelling errors; the posts deal with a lot of different topics, and the point of view is rather subjective. These circumstances make it harder to determine an accurate classification of this kind of text compared to well-written texts, like movie reviews.
Most of the work done in this field deals with tweets written in English, e.g. [2,3]. Since Twitter is also popular in non-English speaking countries, it can be used as a source for informal texts in German language. A use case for such a classification could be a news board displaying the sentiment of the students towards the university or certain classes.

Using the German language in sentiment analysis is considered especially challenging, as the language rules are more complex compared to the English-language, e.g. there is more variety (e.g. »the« vs. »der, die, das«), and nouns must be capitalized. Furthermore sentence syntax is considered more complex in general.

This paper deals with the question whether and how methods of sentiment analysis on English-language tweets can be transferred to German-language tweets. We present a brief overview of related work on English-language texts. Different pre-processing and feature extraction steps are investigated. Four classification algorithms are tested in a complex series of experiments on German tweets. The results are compared to the previous work done on English tweets and it is shown that the feature extraction is one of the most important parts of the whole process in order to gain a high classification quality.

2 Related Work

Many results have been published regarding sentiment analysis on English tweets and movie reviews. This section deals with the results of these papers in more detail.

The first naive attempts of sentiment classification lead one to select two list of words, one corresponding to positive sentiments and one to negative sentiments. This method, as described by Pang et al. in [7] works poorly on movie reviews, at an accuracy of 58 % to 64 %, with the random-choice baseline performing at a 50 % accuracy, since only two equally distributed classes were used. In a second attempt Pang et al. used machine learning methods to improve their results. They used the Support Vector Machine (SVM), Naive Bayes (NB) and Maximum Entropy (ME) algorithms. Furthermore, they used a three-fold cross validation on 700 positive and 700 negative reviews. Their feature vector consists of the frequency or the presence of the most common 16165 uni- and bi-grams. In this case the presence feature vector led to better results. A variety of different methods was tested: uni-grams + bi-grams, bi-grams, uni-grams, part of speech (POS) tagging and more. The best overall performance was achieved with the SVM classifier and the uni-grams presence as feature vector. This approach turned out to be accurate in 82.9 % of the given cases, which was significantly better than the naive approach.

A completely different approach was described by Pang et al. in [6]. The idea here was to check every sentence in the movie review with a machine learning algorithm, like NB or a SVM, to see if the movie review contained a sentiment or was objective. The objective sentences were discarded. The result was a much sparser data set, with almost no information loss. This data set, which only
consisted of two classes, was used as the input of the second stage, in which a graph was built based on the given data. Within this graph a minimum cut was found, which was used to determine the sentiment. Compared to using a simple NB classifier, the accuracy increased from 82.8 % up to 86.4 % for NB used in both steps. The minimum cut approach outperformed the NB and the SVM with 86.4 % (cut) vs. 85.2 % (NB) and 86.15 % (cut) vs. 85.45 % (SVM).

The two above mentioned discussions focused on movie reviews, however Go et al. dealt with Twitter posts as data supply in [2]. They used two sentiments, »positive« and »negative« to classify the tweets, which were written in English. They used the NB, ME and SVM as classifiers and emoticons, like :-) or :-( of the tweets as noisy labels. Their feature vector consists of uni-grams, bi-grams, uni-grams + bi-grams and uni-grams with POS tags. To reduce the corpus size they replaced user names, URL links and repeated letters, like »huuungry«. This reduction led to a size decrease of 45.85 %. The used baseline consists of a word count, in which the words originated from lists for both sentiments. For the uni-gram case the baseline was at 65.2 %; the classifiers result was in NB 81.3 %, ME 80.5 % and SVM 82,2 %. The overall best performance was achieved with uni- and bi-grams and ME with 83.0 % accuracy.

3 Corpus

Since the corpus is extremely important to the performance of the classification task, it is necessary to have a closer look at the corpus which is used. Twitter corpora written in English can be found quite often, but a fully annotated German corpus is harder to find.

There are some German corpora, like the one provided by Bütow et al. in [1], but this corpus only contains news statements, which differ from Twitter statements in various ways. A news text is written more carefully and uses a completely different ductus; furthermore it rarely contains any emoticons, slang words, spelling errors or repeated letters at all.

The corpus provided by Narr et al. [5] contains labeled tweets in multiple languages. For this task we use the German-language part of the corpus, i.e. 1800 German-language tweets labeled as either »positive«, »negative« or »neutral«. This is a difference to the above mentioned discussions [2,6,7], where only »positive« and »negative« classes were utilized. These labels were assigned to the tweets by three different humans via Amazon Mechanical Turk, which means that in some cases the same tweet was labeled differently. To accommodate to this fact, the corpus is split into three data sets. The first set contains all tweets with different labels, now referred to as »agree1«. The second set contains only tweets in which at least two of the labels matched (»agree2«). The last set consists only of tweets in which all three labels match (»agree3«). The »agree2« data set still contains 1719 tweets, while in the »agree3« set only 958 tweets are found. So »agree3« ⊂ »agree2« ⊂ »agree1«. This results in an inter-human-agreement of 1719/1800 = 95.50 % and 958/1800 = 53.22 % respectively. In this corpus some preprocessing has already been done. Links within the tweets
have been replaced with a »~http« tag, also user names have been replaced by »@user«.

This corpus was used in [5] as test-data. Training-data was gained by using another tweet-corpus, labeled using the emoticons as noisy labels. Then a Naive Bayes classifier was used to classify the tweets in this corpus and achieved an accuracy of up to 79.8 % on the German data set.

4  Approach

In this paper we use Python with NLTK and scikit-learn. The resulting script can be obtained under the MIT License from https://bitbucket.org/snippets/mflender/eBKAy.

4.1 Preprocessing

The preprocessing of the tweets contains several steps that are common to natural language processing. These steps will be described in more detail.

The corpus is UTF-8 encoded, since the German alphabet contains some non-ASCII symbols, like ö, ä, ü, ß. However, there are few entries containing non-UTF-8 characters. These lines are filtered first. The next step is to normalize the tweets, i.e. to change the tweets to lower case. This step can be skipped to evaluate the influence of the normalization. To split the tweets into its words and punctuation marks two different tokenizers are used: The WordPunctTokenizer, which treats a simple emoticon as a single word, and the WordTokenizer, which splits an emoticon in its components.

The next step in the process is to remove stopwords. We use a list of German stopwords provided by the NLTK framework. In addition to that list some other characters, like »&«, »)«, »(«, »=« are added to the list of stopwords. To evaluate the influence of stopword removal, this step also can be skipped. As a last preprocessing step the tokens are processed by the Snowball stemmer.

Now the tokens are converted into n-grams. We use several combinations of uni-, bi-, and tri-grams to include word negations and adjectives like nicht gut (not good) or sogar besser als (even better than). An overview of the preprocessing steps and its variations is given in Fig. 1.

Fig. 1. Diagram of the different preprocessing and classification steps
4.2 Feature Extraction

After the pre-processing features are extracted to form the feature vector. To do so two different approaches are used. We refer to the first one as «sparse» and to the second one as «dense». The sparse approach is discussed first:

A list of the \( k \) most common \( n \)-grams in the training set is generated and used as feature vector. All tweets in the data set are represented by an instance of the feature vector, indicating the individual occurrences of the features («binary encoding»). This approach creates a uniform feature space but compared to the dense approach it consumes more memory.

The dense approach features the exact opposite behavior; it uses only a little memory, but creates an asymmetric feature space. For every tweet the feature vector consists of all \( n \)-grams contained in the individual tweet.

A naming scheme to uniquely describe the performed experiments is given in Table 1. This scheme is used to present the results in Table 2, Table 3 and Table 4.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNS</td>
<td>Use cases as they were</td>
</tr>
<tr>
<td>LOW</td>
<td>Set everything to lower case</td>
</tr>
<tr>
<td>WPT</td>
<td>Use the WordPunctTokenizer</td>
</tr>
<tr>
<td>WT</td>
<td>Use the WordTokenizer</td>
</tr>
<tr>
<td>STP</td>
<td>With stopwords and extended stopwords filtered</td>
</tr>
<tr>
<td>nEXSTP</td>
<td>Only with stopwords filtered</td>
</tr>
<tr>
<td>nSTP</td>
<td>Without any stopwords filtered</td>
</tr>
<tr>
<td>STM</td>
<td>With stemmer</td>
</tr>
<tr>
<td>nSTM</td>
<td>Without stemmer</td>
</tr>
<tr>
<td>Uni</td>
<td>Use uni-grams</td>
</tr>
<tr>
<td>Bi</td>
<td>Use bi-grams</td>
</tr>
<tr>
<td>Tri</td>
<td>Use tri-grams</td>
</tr>
<tr>
<td>UniBi</td>
<td>Use uni- and bi-grams</td>
</tr>
<tr>
<td>UniBiTri</td>
<td>Use uni-, bi- and tri-grams</td>
</tr>
<tr>
<td>DNS</td>
<td>Use the dense feature vector</td>
</tr>
<tr>
<td>SPR( (k) )</td>
<td>Use the sparse feature vector with the ( k ) most common ( n )-grams</td>
</tr>
</tbody>
</table>

Table 1. Key to the settings symbols from Table 2, Table 3 and Table 4

4.3 Classification

The process of classification, the classifiers and their settings are now described in more detail. For all classifiers the basic approach is the same. So the general approach is discussed first.

To utilize as much data as possible a ten-fold cross validation is used. The whole corpus is shuffled before classification to prevent influencing the final re-
results by the order of the tweets. The size of the training and test sets, the accuracy of the classification, the time that is used for the training and classification process, the F-measure and its components (precision and recall) are measured for every class. All this information is gained for every fold. The arithmetic mean of all folds is used as the final result.

For the classification four different classifiers are used:

- `sklearn.naive_bayes.MultinomialNB`
- `sklearn.svm.LinearSVC`
- `nltk.DecisionTreeClassifier` and
- `nltk.MaxentClassifier`.

The Multinomial NB and the NLTK decision tree classifier are used without further options. Since NLTK does not provide its own SVM implementation, the Sklearn SVM with linear kernel is used. A pretest of the different kernel settings had shown that the linear kernel performs the best on the given data set. The radial basis function kernel lacks about one percent of accuracy and the polynomial kernel performs even worse.

We use the NLTK implementation of the ME with the MEGAM\(^1\) algorithm. It provides the same accuracy and a higher speed compared to other available ME-algorithms. The number of iterations is set to 20 in a series of pre-experiments. This setting provides the same acceptance rate as the default settings (100 iterations), but takes only a fifth of the time.

Instead of a random-choice baseline, a ZeroR classifier, which decides always for the most common class is used. Thus the percentage occurrences of this class is used as baseline; in the corpus this is the »neutral« one. The used data set is quite small, so it could be possible that in one of the folds there will be no member of the least common class (»negative«). To check upon this possibility we evaluated the proportion of every class in percent for every fold. The result of this test shows that in every fold every class is represented.

5 Evaluation & Results

For the final evaluation of the different classifiers and preprocessing steps only the »agree3« data set is used. The »agree1« and »agree2« data sets are not used to avoid the problem of which sentiment to use in case all the Mechanical Turk workers disagreed. In addition to that, when two or three humans have different opinions on the sentiment of a tweet, that tweet will most likely either have multiple sentiments or its sentiment is very ambiguous.

The »agree3« results are presented in Table 2, Table 3 and Table 4, the explanation for the used acronyms can be found in Table 1. The baseline performance for the experiments is 74.92 % This means that 74.92 % of all tweets in the set were labeled as »neutral«, while 15.07 % where labeled as »positive« and

\(^1\) http://www.umiacs.umd.edu/~hal/megam/
10.01 % as »negative«. This result for the baseline differs slightly from the one computed at [1], due to the filtering of the non-UTF8 lines.

Even without considering variations of parameter $k$, there are about 120 different combinations in the proposed setup, that one can experiment with. These are too many combinations to try out. Therefore, we define a basic setup for both proposed feature vector variants and examine systematically from this scenarios the combination and variation of different parameters and steps to find an optimal setting. Experiment 4 will serve as a standard for the sparse vector and experiment 21 for the dense feature vector. Both standards use all possible preprocessing steps and where tested first. All other variations are employed step by step originating from experiment 4 and 21. For the sparse standard we use $k = 1000$ since it provides a good balance between train + test time and classification accuracy. Kouloumpis et al. also reported this value as useful for sentiment analysis on English-language tweets in [4].

<table>
<thead>
<tr>
<th>Used settings (see Table 1)</th>
<th>NB</th>
<th>SVM</th>
<th>ME</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) SPR(1)</td>
<td>74.92</td>
<td>74.92</td>
<td>74.92</td>
<td>74.92</td>
</tr>
<tr>
<td>(2) SPR(10)</td>
<td>74.92</td>
<td>74.92</td>
<td><strong>75.02</strong></td>
<td>74.92</td>
</tr>
<tr>
<td>(3) SPR(100)</td>
<td>74.92</td>
<td>74.02</td>
<td>74.81</td>
<td><strong>75.13</strong></td>
</tr>
<tr>
<td>(4) SPR(1000)</td>
<td>75.66</td>
<td><strong>78.08</strong></td>
<td>71.97</td>
<td>77.76</td>
</tr>
<tr>
<td>(5) SPR(2000)</td>
<td>70.28</td>
<td>77.34</td>
<td>75.86</td>
<td><strong>77.55</strong></td>
</tr>
<tr>
<td>(6) SPR(3000)</td>
<td>64.17</td>
<td><strong>78.61</strong></td>
<td>75.45</td>
<td>77.13</td>
</tr>
<tr>
<td>(7) SPR(4000)</td>
<td>60.59</td>
<td><strong>79.34</strong></td>
<td>72.81</td>
<td>76.82</td>
</tr>
<tr>
<td>(8) SPR(5000)</td>
<td>60.49</td>
<td><strong>79.87</strong></td>
<td>73.55</td>
<td>77.97</td>
</tr>
<tr>
<td>(9) SPR(10000)</td>
<td>49.11</td>
<td><strong>83.45</strong></td>
<td>75.44</td>
<td>82.82</td>
</tr>
<tr>
<td>(10) SPR(10307)</td>
<td>47.53</td>
<td><strong>83.13</strong></td>
<td>76.92</td>
<td>82.51</td>
</tr>
</tbody>
</table>

Table 2. The results of the $k$ variations for the sparse classification from the »agree3« data set compiled in a table: Accuracy in percent, where the best results per line is written in bold letters

Experiments 1 to 10 of Table 2 show the accuracy affected by the number of features. Experiment 1 only performs with baseline accuracy (74.92 %), since there is not enough information to gain from only one feature. Surprisingly, with just 10 features, the ME classifier is able to perform a little better than baseline. The NB needs more than 100 features to make a decision and its accuracy decreases when more features are available. The corpus contains 10307 distinct features, which is the upper bound for this series of tests.

Experiments 11 to 14 in comparison to experiment 4 show the influence of the different $n$-grams on the accuracy. Most classifiers seem to perform best with the combination of uni- and bi-grams. The exception here is the ME with uni-grams, which performs around 3 % better.

Experiments 4, 15 and 16 show the effect of the stopword removal on the classification. In most cases the accuracy decreases when no stopwords are used. The notable exception here is the ME classifier where the removal of stopwords
slightly (4.59 %) increases the accuracy. Leaving the features without stemming (experiment 17) again seems to reduce the accuracy slightly, with the exception of the ME classifier.

The difference between the two tokenizers can be seen in experiments 18 and 4. Here, the results for the WordPunctTokenizer are slightly better. This is probably due to the fact that the emoticons provide information concerning the sentiment. Experiment 19 and 20, compared respectively to experiment 4 and 10, show that leaving the features in upper and lower case has no significant impact on the accuracy at all. That is interesting, since compared to English, upper and lower case in German is quite important for the spelling of a sentence.

Comparing the standard for the dense feature vector (experiment 21, Table 4) to the sparse feature vector (experiment 4) shows a significant improvement (up to 11 % for the ME) in classification accuracy. With this feature vector only the uni-grams (experiment 22–25) seem to perform well, and it does not make any difference if the uni-grams are paired with bi-grams and/or tri-grams or not.
The overall best result (84.51%) is provided by the ME classifier with removal of the NLTK stopwords (experiment 26). Removing all stopwords (NLTK and extended) decreases the accuracy. Using no stemmer and the WordTokenizer have no positive impact on the resulting accuracy. The effect of leaving the features in upper and lower case is also negligible.

The results from test number 26 with the maximum entropy classifier seems to be the best overall. So this result needs to be investigated further. To do so, we compile the F-measure and its components for every sentiment in Table 5.

<table>
<thead>
<tr>
<th>Neutral</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
<td>F-measure</td>
</tr>
<tr>
<td>0.873</td>
<td>0.949</td>
<td>0.909</td>
</tr>
</tbody>
</table>

**Table 5.** Results from experiment number 26 in greater detail

The data is split into positive, negative and neutral to show how many of the tweets are correctly classified as such. Then for each of these classes the precision, recall and F-measure are listed. The precision shows how many of the tweets classified as a certain class are actually in this class. Recall provides the percentage of all the tweets belonging to this class that was correctly classified as such. Finally, the F-measure is a mixture of precision and recall. The F-measure and recall decrease with the amount of tweets in every class: the neutral class seems to perform best, the negative sentiment performs worst. Maybe this is an indicator, that the ME classifier tends to overfit the data.

As Table 5 shows, all three indicators are especially high for the neutral category. While positive and negative both share similar precision values, the recall for the negative class is especially bad. This is most likely the result of the relatively low number of negative tweets in the corpus. Likewise, the good results for the neutral category can also be explained by the fact, that most tweets in the corpus are considered neutral.

When comparing the classifiers overall, one can see that the SVM and tree classifiers give the best results when using the classical sparse vector, with 83.45 % and 82.82 % (both in experiment 9) accuracy respectively. But when it comes to the dense vectors, the ME classifier gives the best accuracy in this case with 84.51 % (experiment 26) accuracy.

Another crucial characteristic of the text classification is the time it takes to perform a training and test run. Table 6 exemplarily shows some of these values relative to the fastest execution time. The absolute amount of time taken is irrelevant since it depends on the hardware of the used computer. Only experiment 4 and 21 from Table 2 and Table 4 are shown, most of the other execution times are either quite similar or scale predictably, e.g. uni-, bi- and tri-grams take more time than only uni-grams. The dense approach performs with better accuracy and, except for the tree classification, much faster.
Table 6. The train + test time from experiment 4 and 21 from Table 2 and Table 4 in comparison, relative to SVM, experiment 21

6 Discussion

The behavior found here mostly match what has been discovered by Go et al. [2]:

- The best performance of the sparse feature vectors is achieved when the SVM classification is used.
- A combination of \( n \) to \( m \)-grams leads to a better accuracy than only using \( n \)-grams, at least for the sparse feature vector.
- Up to a given point, the amount of used features in the sparse feature vectors lead to a better accuracy.
- The use of preprocessing steps, like stemming and stopword removal, does not seem to have much effect.

Comparing the results of the dense and the sparse feature vectors leads to the assumption, that the classifiers utilize every \( n \)-gram of the dense feature vector and, in doing so, build a model of all available data. In the case of the sparse feature vector in contrast only the \( k \) most common \( n \)-grams may be used, so it ends up with a less accurate model. When the parameter \( k \) has been chosen, so that all \( n \)-grams are utilized, the SVM and tree classifier performance is comparable to the dense feature vector.

To investigate this assumption further, one must have a closer look at the implementation of the decision tree classifier and the SVM. It is supposed that the tree classifier creates the decision tree iteratively from all the given data, so the model it builds is more complete. The SVM skips all the features, which are labeled as «false» in the feature vector, thus effectively reducing the sparse down to the dense feature vector.

The dense feature vector can indirectly learn about the term frequency, since some \( n \)-grams appear more often than others in this feature vector, while every unique \( n \)-gram only appears once for the sparse feature vector. This could also lead to a better overall performance of the dense feature vector.

The difference between German and English (e.g. [2]) text in terms of sentiment analysis is surprisingly small. The stopword removal and stemmer must be changed when working with German text. But despite this, the difference is negligible, which leads to similar accuracies. Most ideas, which have been proven to work in English should easily be adaptable to German. A process of sentiment analysis, that performs well on English-language tweets can, with some minor changes, be transferred to German-language tweets.

Compared to [5] the results presented in this paper are better (79.8 % vs. 84.51 %). However, Narr et al. used a different approach, as our test and train
data originated in the same corpus (using cross-validation), whereas Narr et al. used two different sources.

The ambiguousness of the sentiment classification task is shown by the inter-
human-agreement: Only in 53.22 % of the cases all three humans agreed on the
sentiment, whereas in 95.50 % of all given cases two or more annotators agreed
on the sentiment. Compared to our best result of 84.51 % this displays how
subjective sentiment classification even for real humans is. This fuzziness seems
to provide an upper bound for the classification accuracy.

In comparison to [2, 6, 7] one must notice that the approach presented in this
paper was performed on three different classes instead of two and tweets instead
of movie reviews. Despite these differences the performance of our approach is
still comparable to most of the results provided by Pang et al. and Go et al.

In addition to this one can compare these results with one of the recent deep
learning approaches, e.g. [8]. The best results (85.4 %) Socher et al. gained with
a recursive neural network for two classes are about one percent better than our
best result for three classes (84.51 %). In addition to this one must notice that
Socher et al. used treebank annotations on word level. Since our approach works
with the easier to obtain sentence level annotations and produces a three class
prediction instead of two classes its performance is on the same level as one of
the recent deep learning approaches.

While most of the preprocessing steps have only small influence on the final
results the representation of the feature vector impacts the accuracy and the
time performance significantly.

7 Future Work

In future experiments, some points could be improved. A larger corpus could be
used to gain more reliable results. Another way to improve results would be to
use a better method of selecting features for the sparse vector, like information
gain or $\chi^2$, instead of just using the $k$ most common words. Also one could reduce
the amount of neutral cases within the data set to balance all three cases out and
reduce the bias that comes with the large amount of neutral cases. In addition to
this other classification algorithms like hidden markov models or deep learning
approaches could be used.

Using a bag-of-words approach, structural information is lost. Thus, sentences
containing different sentiments in different parts cannot be detected. Employing
some structure preserving features, e.g. graphs, seems to be promising.

Finally, it appears that there is some kind of upper bound, regarding to what
is accomplishable with classification methods in combination with a bag-of-words
approach. To advance above this level (around 82 % to 84 %), a completely dif-
f erent approach, e.g. as Pang et al. described in [6], might be needed. This upper
bound of around 84 % seems to be independent from the baseline performances,
since Go et al. also reaches this upper bound with a completely different baseline
of 65.2 % [2].
Acknowledgments

We are very grateful to Christian Sander for his help with writing the Python software and running parts of the experiments.

References

MixedTrails: Bayesian Hypothesis Comparison
Heterogeneous Sequential Data

Martin Becker¹, Philipp Singer² Florian Lemmerich²,
Andreas Hotho¹,³, and Markus Strohmaier²,⁴

¹ University of Würzburg, Germany
{becker,hotho}@informatik.uni-wuerzburg.de
² GESIS, Cologne, Germany
{philipp.singer,florian.lemmerich,markus.strohmaier}@gesis.org
³ L3S Research Center, Hannover, Germany
{philipp.singer,markus.strohmaier}@gesis.org
⁴ University of Koblenz-Landau, Mainz, Germany

Sequential traces of user data are frequently observed online and offline, e.g., as sequences of visited websites or as sequences of locations captured by GPS. However, understanding factors explaining the production of sequence data is a challenging task, especially since the data generation is often not homogeneous. For example, navigation behavior might change in different phases of browsing a website, or movement behavior may vary between groups of users. In this work, we tackle this task and propose MixedTrails [1], a Bayesian approach for comparing the plausibility of hypotheses regarding the generative processes of heterogeneous sequence data. Each hypothesis is derived from existing literature, theory or intuition and represents a belief about transition probabilities between a set of states that can vary between groups of observed transitions. For example, when trying to understand human movement in a city, a hypothesis assuming tourists to be more likely to move towards points of interests than locals, can be shown to be more plausible with observed data than a hypothesis assuming the opposite. Our approach incorporates these beliefs as Bayesian priors in a generative mixed transition Markov chain model, and compares their plausibility utilizing Bayes factors. We discuss analytical and approximate inference for calculating the marginal likelihoods for Bayes factors, give guidance on interpreting the results, and illustrate our approach with several experiments on synthetic and empirical data from Wikipedia and Flickr. Thus, this work enables a novel kind of analysis for studying sequential data in many application areas.

References

Predicting Tags for Stack Overflow Questions

Sayan Pal, Ammar Shaker, and Eyke Hüllermeier
Intelligent Systems Group
Paderborn University

Stack Overflow (https://stackoverflow.com/) is one of the major community-driven Question and Answer (Q&A) websites, focusing on topics related to computer programming. It has nearly 7 million users, who ask more than 6700 questions every day. Each question can be associated with up to five different tags, which serve as metadata to facilitate information retrieval.

In this paper, we consider the problem of supporting this process through automatic tagging. From a machine learning point of view, we are facing a problem of Extreme Multi-Label Classification (XMLC), as Stack Overflow allows for choosing from several thousands of tags. Besides, instead of learning in a standard batch mode, it is desirable to learn incrementally on the continuous stream of questions entering the system, with the capability to capture changes and drifts in the data: for example, many tags (such as ‘facebook’) have a lifetime, first gaining popularity, then reaching a peak and eventually diminishing.

Thus, we end up with an extremely challenging problem of XMLC for data streams with a non-stationary set of labels. To tackle this problem, we build on an XMLC method based on probabilistic label trees (PLT), which has recently been proposed in [1]. We extend this approach in two directions. First, instead of specifying the entire PLT beforehand, we develop an adaptive version that starts with only a single node and expands the tree whenever a new label is observed in a training example. As a second contribution, we further improve adaptive PLTs through stream-based boosting [2]. More specifically, we apply the online boosting method by Oza and Russell, which we tailor for minimizing the F-measure as a performance metric (instead of 0/1 loss, which is not appropriate in the context of XMLC).

In addition to the methodological contributions, we present empirical results based on extensive experiments with real data from Stack Overflow. Our experimental setting is focused on evaluating the usefulness of the extensions we proposed for PLTs, i.e., the adaptive handling of labels and online boosting.

References
Finding Hierarchy of Topics from Twitter Data

Nghia Duong-Trung, Nicolas Schilling, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Universitätspalat 1, 31141 Hildesheim, Germany
{duongn,schilling,schmidt-thieme}@ismll.uni-hildesheim.de
http://www.ismll.uni-hildesheim.de

Abstract. Topic modeling of text collections is rapidly gaining importance for a wide variety of applications including information retrieval and automatic multimedia indexing. Our motivation is to exploit a hierarchical topic selection via nonnegative matrix factorization to capture the nature content of text posted on Twitter. This paper explores the use of an effective framework to automatically discover hidden topics and their sub-topics. As input, the framework uses textual data. The output is then the discovered structure of topics. We introduce a conceptual topic modeling based on the idea of stability analysis to detect a hierarchy of topics given a text source. In this process, we apply stability measurement in conjunction with nonnegative matrix factorization and WordNet to excavate hidden topics by the scores of conceptual similarity. To demonstrate the effectiveness and generalization, we apply the approach to a large-scale Twitter dataset to investigate the content topics. We also address the problems of several state-of-the-art topic modeling approaches that are unable to handle a large dataset.

Keywords: Unsupervised Learning, Semantics in Text Mining, Conceptual Stability, Hierarchy of Topics

1 Introduction

Nonnegative matrix factorization (NMF) with nonnegativity constraints has been considered as an efficient representation and an emerged technique for text mining and document clustering [22,2,17,23,9,11]. For any desired low-rank $K$, the NMF algorithm groups the data into clusters. The key issue is whether a given low-rank $K$ helps to decompose the data into appropriate separated clusters. Therefore, the problem we study in this paper is how to effectively and efficiently discover the most appropriate structure of topics giving a text corpus by exploiting the semantic meaning and the conceptual stability. In general, the stability of a clustering model refers to its ability to consistently replicate similar solutions on data randomly generated from the same source. In practice, this involves a repeated re-sampling of data, applying a topic selection model, and evaluating the results by a stability metric which measures the level of discrimination between the resulting clusterings.
We start with the previous work on the idea of random sub-sampling and stability analysis via consensus clustering to discover the number of clusters that best describes the data [16,4,13]. The basic assumption of stability in the context of consensus clustering, in general, is very intuitive: for particular observed data, if we perturb it into different random variabilities, and if they produce the same cluster composition, or consensus, without radical difference, we would confidently consider that these clusters represent real structure. Consensus clustering purely captures this procedure. Further work investigated by [4] improved the consensus clustering technique by adding a quantitative evaluation for robustness of the decomposition. They adopted a measure based on the cophenetic correlation coefficient which indicates the dispersion of the consensus matrix. The coefficient is calculated as the Pearson correlation of two distance matrices: the consensus matrix captured the distance between data samples and the average connectivity matrix over many clustering runs. Subsequently, [12,13] formulate the idea of consensus matrix in the latent space learned by NMF.

However, the computation of consensus matrix, $R^{n \times n}$ matrix where $n$ is the number of tweets/documents, seems very costly, e.g. large amount of RAM is required. For instance, if we apply the previous method on our experimented Twitter dataset that we describe later in the paper, then 1400GB of RAM is required to store the consensus matrix during model’s computation. Hence, the method provided by [12,13] is insufficient or even impossible for large datasets. To overcome the drawbacks of the construction of consensus matrix, we propose a topic selection approach, called the conceptual stability analysis, to smoothly integrate with NMF that can be applied on large datasets effectively.

Moreover, we also evaluate several state-of-the-art topic modeling approaches via Latent Dirichlet Allocation (LDA) [3]. The first baseline is the topic selection method implemented by [1]. The second baseline is proposed by [6]. We implement the baselines by using [10,18]. However, these methods threw exception due to large dataset during computation. An upper bound of RAM required for each approach is 65GB until an exception occurs.

With these limitation in mind, we introduce an unsupervised topic selection method that enhances the accuracy and effectiveness of NMF-based models in the context of document clustering and topic modeling. We show that our proposed method can work effectively on large dataset within acceptable computing resources such as RAM required and time of computation.

2 Theoretical Aspects and Proposed Framework

2.1 Nonnegative Matrix Factorization

Consider a dataset $X \in \mathbb{R}^{n \times m}$ containing a set of $n$ documents where each document is described by $m$ many features. The document features are mapped from a dictionary that comprises all words/terms/tokens in the dataset. Each positive entry $X_{ij}$ is either a raw term frequency or a term frequency - inverse document frequency (TFIDF) score. By $r$ and $\tau$, we denote the sampling rate
and the number of subsets generated from $X$ respectively. Then each subset $X_\tau \in \mathbb{R}^{n\times m}$ is a sample without replacement of $X$.

Giving a desired number of topics $k$, the NMF algorithm iteratively computes an approximation:

$$X \approx WH,$$

where $W \in \mathbb{R}^{n\times k}$ and $H \in \mathbb{R}^{k\times m}$ are nonnegative matrices. The conventional technique to approximate $W$ and $H$ is by minimizing the difference between $X$ and $WH$ such that:

$$\min_{W \geq 0, H \geq 0} f(W, H) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} (X_{ij} - (WH)_{ij})^2 + \phi(W) + \theta(H),$$

where $\phi(.)$ and $\theta(.)$ are regularization terms that are set as follows:

$$\phi(W) = \alpha \|W\|_F^2 \quad \text{and} \quad \theta(H) = \beta \sum_{i=1}^{m} \|H(:,i)\|_1^2,$$

where $H(:,i)$ indicates the $i$-th column of $H$. The $L_1$ norm term of $\theta(H)$ promotes sparsity on the rows of $H$ while the Frobenius norm term of $\phi(W)$ prevents $W$ from growing too large. Scalar parameters $\alpha$ and $\beta$ are used to control the strength of regularization. The matrices $W$ and $H$ are found by minimizing Equation (2) via estimating $W$ and $H$ in an alternating fashion using projected gradients or coordinate descent [5].

### Table 1. Summary of topics discovered.

<table>
<thead>
<tr>
<th>N</th>
<th>Topic</th>
<th>Number of sub-topics</th>
<th>Number of documents</th>
<th>Share (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student Life and Relationship</td>
<td>5</td>
<td>76,415</td>
<td>17.78</td>
</tr>
<tr>
<td>2</td>
<td>Information and Networking</td>
<td>8</td>
<td>17,649</td>
<td>4.11</td>
</tr>
<tr>
<td>3</td>
<td>Business and Current Affairs</td>
<td>2</td>
<td>72,534</td>
<td>16.88</td>
</tr>
<tr>
<td>4</td>
<td>Routine Activities</td>
<td>4</td>
<td>21,960</td>
<td>5.11</td>
</tr>
<tr>
<td>5</td>
<td>Leisure and Entertainment</td>
<td>2</td>
<td>81,469</td>
<td>18.96</td>
</tr>
<tr>
<td>6</td>
<td>Sport and Games</td>
<td>3</td>
<td>31,812</td>
<td>7.40</td>
</tr>
<tr>
<td>7</td>
<td>Pessimism and Negativity</td>
<td>4</td>
<td>78,329</td>
<td>18.23</td>
</tr>
<tr>
<td>8</td>
<td>Wishes and Gratitude</td>
<td>5</td>
<td>36,618</td>
<td>8.52</td>
</tr>
<tr>
<td>9</td>
<td>Transport and Travel</td>
<td>2</td>
<td>12,887</td>
<td>3.01</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td><strong>429,673</strong></td>
<td><strong>100.00</strong></td>
</tr>
</tbody>
</table>

### 2.2 Conceptual Stability Computation

Now we start discussing our approach of computing stability based on the usage of the WordNet hypernym hierarchy [8,15]. Given tokens $c_p$ and $c_q$, then
Table 2. Labels of all 9 topics and 35 sub-topics.

<table>
<thead>
<tr>
<th></th>
<th>Student Life and Relationship</th>
<th>Information and Networking</th>
<th>Routine Activities</th>
<th>Pessimism and Negativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Friends and Relationship</td>
<td>2.1 News</td>
<td>4.1 Feelings</td>
<td>7.1 Hate and Anger</td>
</tr>
<tr>
<td>1.2</td>
<td>Study Life</td>
<td>2.2 Life</td>
<td>4.2 Sleep</td>
<td>7.2 Daily Problems and Complains</td>
</tr>
<tr>
<td>1.3</td>
<td>Worry and Confusion</td>
<td>2.3 Mood and Reflections</td>
<td>4.3 Work and People</td>
<td>7.3 School Routines</td>
</tr>
<tr>
<td>1.4</td>
<td>Conversations</td>
<td>2.4 Greetings</td>
<td>4.4 Social Media</td>
<td>7.4 Life and Changes</td>
</tr>
<tr>
<td>1.5</td>
<td>Social Media and Connections</td>
<td>2.5 Current Regional Events</td>
<td>5 Leisure and Entertainment</td>
<td>9 Transport and Traveling</td>
</tr>
<tr>
<td>8</td>
<td>Wishes and Gratitude</td>
<td>2.6 Mates</td>
<td>5.1 Television and Cinema</td>
<td>9.1 Landmarks</td>
</tr>
<tr>
<td>8.1</td>
<td>Friends</td>
<td>2.7 Informal Chat</td>
<td>5.2 Reactions</td>
<td>9.2 Journeys</td>
</tr>
<tr>
<td>8.2</td>
<td>People</td>
<td>2.8 Religion</td>
<td>6 Sport and Games</td>
<td></td>
</tr>
<tr>
<td>8.3</td>
<td>Anticipation</td>
<td>3 Business and Current Affairs</td>
<td>6.1 Sport Opinions and Discussion</td>
<td></td>
</tr>
<tr>
<td>8.4</td>
<td>Thanks and Affection</td>
<td>3.1 Working-day Activities</td>
<td>6.2 American Football</td>
<td></td>
</tr>
<tr>
<td>8.5</td>
<td>Celebrations</td>
<td>3.2 Events and Socializing</td>
<td>6.3 TV Sport Programs</td>
<td></td>
</tr>
</tbody>
</table>

\[
wup(c_p, c_q) = \frac{2d}{d_1 + d_2 + 2d}
\]  

(4)

where \(d_1\) and \(d_2\) are the distances that separates the concept \(c_p\) and \(c_q\) from their closest common ancestor and \(d\) is the distance which separates the closest common ancestor of \(c_p\) and \(c_q\) from the root node.

Each row of the low-rank matrix \(H\) represents one of the \(k\) topics and consists of scores for each term. However, we only consider the top \(t \ll m\) terms as they contribute most to the semantic meaning of a topic. In practice, the contribution of each token to topic \(i\) is represented by the scores in the \(i\)-th row in matrix \(H\) generated by NMF. By sorting each row of \(H\), we can assess the top \(t\) terms for each topic. The set of top \(t\) tokens for all topics of a given \(H\) will be denoted by \(S = \{R_1, \ldots, R_k\}\) such that \(R_i \in \mathbb{R}^t\) is the topic \(i\)-th represented by top \(t\) tokens. Within a topic, we calculate the conceptual stability score as follows:

\[
sim(R_i) = \frac{2}{t(t-1)} \sum_{i=0}^{t-1} \sum_{j=i+1}^{t} wup(R_{vi}, R_{vj})
\]

(5)

Similarly, the conceptual stability score between two topics \(R_u\) and \(R_v\) is calculated in the same fashion.

\[
sim(R_u, R_v) = \frac{1}{t^2} \sum_{i=1}^{t} \sum_{j=1}^{t} wup(R_{ui}, R_{vj})
\]

(6)
Finally, we consider the problem of measuring the conceptual stability between two different $K$-way topic clusterings $S_w$ and $S_l$. Each ranked list contains top $t$ tokens that contribute most semantic meaning to the $i$-th topic. Then, the conceptual stability between $S_w$ and $S_l$ is calculated by:

$$
\text{con}(S_w, S_l) = \frac{1}{K} \sum_{k=1}^{K} \sim(R_{wk}, \pi(R_{lk}))\),
$$

where $\pi(R_{wi})$ denotes the ranked list $R_{lj}$ matched to the ranked list $R_{wi}$ by the permutation $\pi$. The optimal permutation $\pi$ is found by solving the minimal weight bipartite matching problem using the Hungarian method [14].

Moreover, the problem of measuring the conceptual stability within the $K$-way topic clustering $S_w$ itself is also considered. The conceptual stability is then calculated as follows:

$$
\text{con}(S_w) = \frac{1}{K} \sum_{k=1}^{K} \sim(R_{wk})
$$

We now consider the conceptual stability at a particular number of topics $k$. At first we apply the NMF on the complete dataset $X$ to get the factor matrices $H$ that we consider as the reference ranked lists. Let us define $S_X$ as the reference $K$-way topic clustering, containing $K$ ranked lists $S_X = \{R_{X1}, \ldots, R_{Xk}\}$.

Subsequently, we randomly resample $\tau$ times the documents of the original $X$ with the sampling rate $r$ to obtain a random subset of $X$ which we denote by $X_\tau$. We then apply NMF on each $X_\tau$ to get the factor matrix $H_\tau$. This results in $\tau$ many sets $\{S_1, \ldots, S_\tau\}$ where each set contains $k$ ranked lists $S_j = \{R_{j1}, \ldots, R_{jk}\}$. Finally, we calculate the overall semantically conceptual stability at $k$ as following:

$$
\text{stability}(k) = \frac{1}{\tau} \left| \sum_{i=1}^{\tau} \text{con}(S_X, S_i) - \sum_{i=1}^{\tau} \text{con}(S_i) \right| \max(\sum_{i=1}^{\tau} \text{con}(S_X, S_i), \sum_{i=1}^{\tau} \text{con}(S_i))
$$

The maximum stability score is achieved if and only if the top $t$ tokens appear in only one topic $k$. Otherwise, the minimum stability score is obtained if top $t$ tokens overpoweringly appear in every topic $k$.

This process is repeated for a range of topics $k$. The most appropriate value of $k$ is identified by the highest value of stability($k$) score. However, the scores also reveal the possible range of $k$ for further investigation. With the $k$ topic classification finalized at the first level, the dataset is split into sub-datasets where documents are assigned to topic with the highest score, e.g. through the $W$ matrix.

$$
\hat{k}_{Xi} = \arg\max_k(W_{ik})
$$

Then, the process is repeated to discover sub-topics in each sub-dataset. Generally, we can expand the procedure deeper in the hierarchy. First, we calculate the most appropriate number of topics $L$ in the whole dataset $X$. Then, a subset of $X$ is drawn based on each value in range $k \in l, \ldots, L$. The stability($k$) score is, in turn, calculated for each subset to find the best number of sub-topics.
Algorithm 1 The conceptual stability analysis approach with 2-level of hierarchy

**Input:** Dataset $X \in \mathbb{R}^{n \times m}$, range of number of topics $[K', \ldots, K'']$, number of top tokens $t$, sampling rate $r$, number of subsets $\tau$

1: /* find $k$ at the first level in hierarchy */
2: for $k \in K', \ldots, K''$ do
3: find $W \in \mathbb{R}^{n \times k}, H \in \mathbb{R}^{k \times m}$ with $X \approx WH$
4: get $S_X \in \mathbb{R}^{k \times t}$ from $H$
5: for $\tau \in 1, \ldots, \tau$ do
6: draw $X_\tau \in \mathbb{R}^{n' \times m}$ from $X$
7: find $W_\tau \in \mathbb{R}^{n' \times k}, H_\tau \in \mathbb{R}^{k \times m}$ with $X_\tau \approx W_\tau H_\tau$
8: get $S_{X_\tau} \in \mathbb{R}^{k \times t}$ from $H_\tau$
9: end for
10: calculate $\text{stability}(k)$, Equation (9)
11: end for
12: $L = \text{argmax}_k \text{stability}(k)$
13: /* find $k$ at the second level in hierarchy */
14: for $h \in 1, \ldots, L$ do
15: $X_h = \emptyset$
16: for $X_i$ s.t. $h = \text{argmax}(W_{ih})$ do
17: $X_h = X_h \cup \{X_i\}$
18: end for
19: for $k \in K', \ldots, K''$ do
20: find $W_h \in \mathbb{R}^{p \times k}, H_h \in \mathbb{R}^{k \times m}$ with $X_h \approx W_h H_h$
21: get $S_{X_h} \in \mathbb{R}^{k \times t}$ from $H_h$
22: for $\tau \in 1, \ldots, \tau$ do
23: draw $X_{h,\tau} \in \mathbb{R}^{p' \times m}$ from $X_h$
24: find $W_{h,\tau} \in \mathbb{R}^{p' \times k}, H_{h,\tau} \in \mathbb{R}^{k \times m}$ with $X_{h,\tau} \approx W_{h,\tau} H_{h,\tau}$
25: get $S_{X_{h,\tau}} \in \mathbb{R}^{k \times t}$ from $H_{h,\tau}$
26: end for
27: calculate $\text{stability}(k)$ as Equation (9)
28: end for
29: $L_h = \text{argmax}_k \text{stability}(k)$
30: end for

For the ease of interpretation, we conduct the experiments within 2-level of hierarchy. An overview of the whole procedure can be seen in Algorithm (1).

3 Empirical Results

3.1 Datasets, Experiment Setup, and Baselines

The North America dataset is a large dataset of tweets that was originally used for the geolocation prediction problem [19,20,7]. A document in this dataset is
Fig. 1. Experiment results on the Tweets dataset. Figure (1(a)) shows discovered topics at the first level. Similarly, the other figures present discovered topics at the second level. The appropriate number of topics $k$ are identified by peaks in the plots. The vertical lines represent the highest peaks $k$.

The concatenation of all tweets by a single user. There were total 38 million tweets tweeted by 430k users. The tweets were inside a bounding box covering the continuous United States, a part of Canada and a part of Mexico. The final
dataset after preprocessing is a very sparse text file that requires 2.2GB to store and contains 430,000 rows, e.g. the number of documents, and 59,368 columns, e.g. the vocabulary size.

In our experiment, we set required model parameters as follows. The range of exploring topics at the first level is \{5, \ldots, 25\}. We expect the range of sub-topics is smaller in the second level so that the range of exploring sub-topics is \{2, \ldots, 12\}. The number of top tokens that characterize a specific topic is set to \( t = 20 \). The sampling rate is set to \( r = 0.8 \) and the number of subsets is set to \( \tau = 25 \) to cancel out random effects. Our experiments were conducted on a Xeon E5-2670v2 with 2.5GHz clock speed and 128GB of RAM. However, an upper bound of RAM required for our model is 5GB and it takes 4 days to complete.

As we already mentioned in the introduction section, during our experiment, we also compare our method with several state-of-the-art NMF-based and LDA-based topic modeling approaches [13,1,6]. However, all these models either cannot handle a large dataset or throw resource exception during computation.

Fig. 2. Experiment results on discovered topics at the second level. The appropriate number of topics \( k \) are identified by peaks in the plots. The vertical lines represent the highest peaks \( k \).
3.2 Topics Discovery

The framework identifies distinctive topics and their sub-topics of documents based on the output of stability scores. In theory, the deepest hierarchy of topics is where documents are recursively classified until one topic only contains one document. We do not specify the exact number of topics beforehand but rather the range of desired topics and the model will figure out the most appropriate values itself. In other words, the model takes (1) a very large textual dataset, (2) a desired range of expectant number of topics, and (3) a desired level of hierarchy. Then, the hierarchy of topics is discovered by considering conceptual stability scores.

Figure (1,2) present the potential number of topics and sub-topics at the first and second levels respectively. Table (1) summarizes topics and their sub-topics explored. As we can see in Table (1), topics at the first level can be divided into two groups based on the % share. People are concerned the most about Pessimism and Negativity, Leisure and Entertainment, Student Life and Relationship and Business and Current Affairs. We now describe all the topics and their sub-topics discovery in more detail.

At the first level, the highest peak is found at \( k = 9 \) which means that the most distinctive number of topics given North America tweet dataset is 9. However, we also see potential high peaks at \( k = 11 \) and \( k = 7 \) if we need manually to expand or condense the clustering results respectively. Consequently, the whole dataset at the first level is then divided into 9 sub-datasets that the model continues discovering sub-topics within them.

Next we consider sub-topics. Figure (1(b)) presents that the highest peak is at \( k = 5 \) where we clearly see a \( \Lambda \) shape. Similarly, we see the same \( \Lambda \) shape in the 4\(^{th}\), 6\(^{th}\), 7\(^{th}\) and 8\(^{th}\) topics which are presented in Figure (1(e), 2(a), 2(b) and 2(c)) respectively. The peaks made by the \( \Lambda \) shape is the number of sub-topics discovered by the model. Interestingly, the 2\(^{nd}\) topic, Figure (1(c)), contains only 4.11% of the documents but can be divided into \( k = 8 \) distinctive sub-topics. The 3\(^{rd}\) and 9\(^{th}\) topics, Figure (1(d), 2(d)) respectively, show an obvious peak that the most suitable number of sub-topics is \( k = 2 \), the left most bound of the experimented range. The 5\(^{th}\) topic, Figure (1(f)), presents two candidates with high magnitude peaks at \( k = 2 \) and \( k = 7 \). Although the highest peak is selected, e.g. \( k = 2 \), as the output for sub-topics consideration, user can manually choose the other peak as the desired output.

3.3 Topics Labeling

Having exploited the hierarchical topics structure, we next present their associated labels. Table (2) summarize our labeling schemes. All topics and sub-topics were subjectively labeled to ease the understanding and interpretation in successive spatial distribution analysis. The labels were validated and assigned based on the meaning of top tokens that characterize a specific topic or sub-topic.

More generally, questions of accuracy can be raised about the representativeness of labels as a source for topics demonstration. In each discovered topic
and sub-topic, after collecting top tokens based on their meaning contribution, a wide number of heuristic labeling schemes is considered to render each topic representative and distinctive. After the labels are generated, a random selected documents are reviewed and the labels are re-validated if needed. The loop is required to ensure the assigned labels are acceptably appropriate. It is important to consider that the labeling results from this paper reflect Twitter users’ opinions at the time the data was collected, not the population at large. The revealed Twitter topics also were visualized using a comparison word cloud of the top tokens in all topics and sub-topics, e.g. Figure (3). We report the principal component analysis to inspect the subjective distinctiveness of topics in Figure (4).

4 Conclusion

In this paper, we propose a topic selection approach to smoothly integrate with NMF that can be applied on large datasets effectively. The model automatically discovers the most distinctive topics and sub-topics in many levels of desired hierarchy by considering conceptual stability scores. The conceptual analysis helps guide the selection of the appropriate number of topics and their sub-topics. The main strength of our approach is that it is entirely unsupervised and does not require any training step. We also demonstrate the practicability of our framework to get a better understanding of textual source. Starting from
Fig. 4. The bubble plot of 35 discovered sub-topics. The size of the bubbles corresponds with the % share assigned to that sub-topic. For the ease of interpretation, we report the bubble plot as 1-component principle component analysis.

addressing the drawbacks of consensus matrix models that exist more than a decade, we have provided an effective and powerful framework for large-scale text mining and document clustering via NMF. We also present several state-of-the-art LDA-based topic modeling approaches that are unable to handle large dataset.

References

Using Clustering for Categorization of Support Tickets

Daniel Beneker and Carsten Gips

FH Bielefeld - Campus Minden,
Artilleristraße 9, 32427 Minden, Germany

http://www.fh-bielefeld.de/

Abstract. Support tickets from customers contain much hidden information. Unsupervised machine learning methods are able to discover this hidden information. In this paper we propose the categorization of support tickets using clustering methods in combination with topic models. Furthermore label generation techniques are used to generate meaningful names for these categories. The results are compared with related research.

Keywords: clustering, support tickets, categorization, topic model, nmf, k-means, information retrieval

1 Introduction

The volume of digital data generated annually has grown massively in recent years, but only a small amount of this data is used for analysis. Manual analysis of these data sets is not only more difficult, but also more cost-intensive. Therefore, it is important to organize and categorize them appropriately. These tasks can be successfully automated with clustering algorithms.

At Mittwald CM Service GmbH & Co. KG a large part of the communication with the customer takes place over the ticket system of the company. The customer makes a new request via the ticket system, the so-called support ticket. This is handled by employees, so that usually a conversation sequence consisting of several internal and external tickets is created. External tickets are tickets from the customer as well as responses of the employees to the customer. Internal tickets contain additional information and are only visible to employees.

Since these tickets are currently not analyzed, it is difficult to evaluate for which topic the company receives the most requests. This paper presents an approach to cluster support tickets into categories. The algorithms k-means and NMF are used for this purpose, which belong to the field of unsupervised learning and do not require labelled training data. In a second step label generation techniques are used to generate meaningful names for the calculated categories. These meaningful names are important because, without them, it is not possible to evaluate which topics cost how much time or how much money. With the help of the named categories, it is also possible that an employee, who is an expert in a particular topic, processes the corresponding tickets prioritized.
In Section 2 we briefly discuss different approaches for clustering documents and present in Section 3 our approach to cluster the support tickets and to extract meaningful names for the found clusters. The results are evaluated in Section 4.

2 Related Work

There are many different approaches to categorize texts and documents. In [1] Blei uses the LDA (Latent Dirichlet Allocation) algorithm to categorize 17,000 articles from the journal *Science* into 100 categories. The found categories provided a capable and general overview of the individual subjects. LDA belongs to the topic models as well as the algorithm NMF used in this work. An extension to LDA is described in [9]. The hierarchical variant presented there was used to categorize news texts around the disappearance of the Malaysia airline flight *mh-370*. As resulting categories the individual theories about the status of the plane like *plane crash* or *terrorism* were obtained.

In [4], Kuang et al. show how NMF (Nonnegative Matrix Factorization) can be used to classify documents. They also test NMF on five different data sets like the well-known sets *20Newsgroups* and *Reuters* and compare the results to a baseline k-means clustering. In four out of five cases, NMF yielded better results. In [5] Shahnaz et al. note, that the accuracy of the clustering strongly depends on the number of clusters and the dataset. With only 2 categories of the *Reuters* dataset, an accuracy of 99% was achieved. With 20 categories it is only 54%. But on another dataset, they received over 80% accuracy with 20 categories.

Other approaches attempt to generate training data in a first step using cluster algorithms and keyword-lists, and to train a supervised classifier such as SVM or Naive Bayes in a second step. The classifier is then used to categorize documents. So they combine unsupervised learning methods with methods of supervised learning. In [2] Ko and Seo split the documents into sentences and categorize them by a list of keywords. Those sentences are used to train a naive bayes classifier. The results did not differ significantly from those of pure supervised learning. But the approach is helpful when training data must be generated. An approach that uses automatic created keywords to create training data is described in [3]. They evaluate their approach on the *20Newsgroups* and *Reuters* dataset and achieved the best results using only a small number of keyword. Furthermore, they received a small improvement, if humans filter those automatic created keywords.

The algorithm DCF (Description Comes First) is described in [10]. That approach is different from the others. First, label candidates are generated for each text. Then the texts are categorized (for example with k-means). The final step the centroids are used to select one of the label candidates.

3 Approach

In order to categorize support tickets, the tickets are preprocessed and converted into feature vectors, and cluster algorithms are applied for categorization. In the
last step meaningful names are assigned to the categories (see Figure 1). After initial clustering, the trained algorithm can be saved to get fixed categories and to predict the category of new tickets later.

Fig. 1: Approach Overview

3.1 Preprocessing

This section describes how the support tickets are preprocessed so that they can be used by algorithms such as k-means and NMF.

**Normalization** A support ticket consists of two parts, the subject and the message. For further processing subject and message are concatenated and converted to lower case. Support tickets can contain many elements that can adversely affect the quality of clustering. After loading the tickets, elements like URLs, e-mails and account or database names are removed using regular expressions. Single numbers are also removed, as we want strings as categories.

**Stopword removal** Stopwords contain little or no information and can therefore be removed. We used the stopword list from the NLTK framework[1] extended by a list containing words which could have a negative impact on the automatic naming of the clusters, such as salutation and greeting formulas. Especially short tickets could otherwise be assigned to a common cluster called freundliche Grüße.

**Text tokenization** To split the normalized texts into token lists we used a simple tokenizer with the following regular expression: `/\w\w\w+/u`, to capture all words of length 3 and longer. Other characters such as punctuation are discarded, since the cluster labels should not contain punctuation marks.

**Generation of feature vectors** To cluster texts using algorithms, they must first be converted into numbers. One way is to use the *term frequency - inverse document frequency* (*tf-idf*) vectors. These feature vectors are based on two assumptions. First, words that are more common in a document are more important, or they describe the document better, and second, words that are common in all documents are unimportant. The tf-idf value of a word \( t \) in a document \( d \) is calculated as follows:

\[
\text{tf-idf}_{t,d} = tf_{t,d} \times idf_t
\]  

(1)

\( tf_{t,d} \) specifies the occurrences of a word \( t \) in a document \( d \). This value is then weighted with the term \( idf_t \). This is calculated as follows:

\[
idf_t = \log \left( \frac{N}{df_t} + 1 \right)
\]  

(2)

\( N \) is the number of all documents. \( df_t \) stands for *document frequency* and specifies in how many documents the word \( t \) occurs. For rarely used words, the \( idf_t \) value is large; for common words the value is small.

After this step, each document is represented by a feature vector with the length of the vocabulary. At this point, if desired, all words above and below a certain occurrence frequency can be specifically excluded in order to reduce the length of the feature vectors. This has the advantage that, for example, words with spelling mistakes, which occur very rarely, are filtered out.

### 3.2 Algorithms

We use k-means because it is a simple baseline algorithm and NMF because it belongs to the topic models and provides descriptive results as we want to compare a standard cluster algorithm and a topic model.

**k-means** K-means represents a simple algorithm to classify data into clusters. It can be divided into 3 steps. In the first step, the cluster centroids \( \{c_1, \cdots, c_K\} \) are initialized for randomly. In the second step, each data point \( x \) is assigned to the cluster with the smallest distance. Following, the cluster centroids are recalculated in a third step. The centroid is always calculated from the average of all the data points assigned to it. Steps 2 and 3 are repeated until cluster centroids no longer change.

**Non-negative Matrix Factorization (NMF)** Non-negative matrix factorization describes a method in which a matrix is represented by a product of two smaller matrices. This method can also be used to cluster documents.\[5\]

Given a matrix \( V \), it can be decomposed with NMF into the matrices \( W \) and \( H \), so that \( W \times H \approx V \). If \( V \) is the size \( m \times n \), then \( W \) has the size \( m \times k \) and \( H \)
the size $k \times n$. The goal is that the product of $W$ and $H$ should match the original matrix $V$. From this the following minimization problem can be derived: [8]

$$\min_{W, H} \| V - WH \|_F^2$$

(3)

This minimization problem can be solved by a special gradient descent method as described by [6]. $\| \cdot \|_F$ stands for the Frobenius norm. It specifies the size of a matrix and is defined as:

$$\| K \|_F = \sqrt{\sum_i \sum_j |k_{ij}|^2}$$

(4)

This method can also be used for clusters of documents. Figure 2 shows what information the matrices $V$, $W$, and $H$ contain. The matrix $V$ contains all the documents to be used for clustering. A document is always represented by a number vector of length $n$. $n$ corresponds to the length of the vocabulary over all documents in $V$. As a rule, vectors with weighted word frequencies, such as $tf-idf$, are used. $m$ stands for the number of documents. An entry $V_{mn}$ from $V$ thus contains a value that reflects the meaning of the word $n$ in the document $m$ (if $tf-idf$ is used). $K$ in the smaller matrices $W$ and $H$ stands for the number of clusters. As with k-means, this value must be set by the user beforehand. After decomposing $V$ into the matrices $W$ and $H$, $W$ contains the assignments of the individual documents to the clusters. The largest number in each line decides about the belonging to a cluster. $H$ contains a probability distribution of words over clusters. The larger the entry $H_{ig}$ in $H$, the more likely it is that the word $g$ occurs in the documents from cluster $i$. The matrix $H$ can be used to generate names or descriptions for the respective clusters. [8]

NMF can be assigned to the topic models rather than to the classical cluster algorithms, such as k-means. Topic Model, similar to cluster algorithms, can find patterns in documents. They do not split the documents into disjoint clusters, but into so-called topics, that is, Clusters that overlap. For example, a document can belong to 90% of Topic 1, to 8% of Topic 3, and 2% of Topic 2. The subdivision into disjoint clusters with a topic model can be done by assigning a document only to the topic with the highest value. [1]

3.3 Label generation

Without descriptive cluster names, it is difficult for humans to gather information from the grouped documents. One of the most difficult tasks in document clustering is therefore to generate a description which is understandable to humans. For algorithms from the field of supervised learning this is easily possible via the training data. However, these are not present during clustering.

The naming of clusters is often referred to as cluster labeling in the literature and can be done in many different ways. They can be divided into manual, automatic and semi-automatic approaches. The manual naming of clusters is done by the human being who creates a name or description after visual inspection of
some documents of a cluster. Automatic methods attempt to generate descriptions from the most powerful features within a cluster. Semi-automatic approaches combine these two possibilities, for example, by customizing the automatic descriptions by the human being. Another possibility is that a human describe categories with lists of keywords. Subsequently, the documents are automatically sorted into these categories. Other approaches use additional external information, such as the WordNet database, which provides lexical-semantic relationships between words. 

The implemented solution within the scope of this work consists of an automatic generation of labels and offers the option to adjust the labels manually afterwards. Figure 3 introduces the individual steps, which are described in more detail below.

**Extraction of label candidates** The first step of cluster labeling is to extract label candidates. For each cluster, all words in the vocabulary are sorted according their relevance to the cluster. When using NMF to cluster the documents, the \( H \) matrix already contains all the information you need. For each word and cluster, the \( H_{ig} \) in \( H \) entry contains a value that reflects the relevance of the word \( g \) in cluster \( i \) (see section 3.2). It is therefore sufficient to sort the matrix \( H \) line by line. The larger a value, the more important is the corresponding word.

For k-means as a clustering algorithm, this information can be obtained via the cluster centroids. The cluster centroids are located in the feature space and therefore have the same dimension as the tf-idf feature vectors. In order to obtain
a list of words sorted by relevance for k-means, the values of the cluster centroids must be sorted.

Figure 4 shows the matrix $H$ with three different feature vectors. They each contain the two features dog and cat. For example, to obtain all label candidates for cluster 1, the values of the row are sorted descending. On the right, the image shows the characteristic space with the found cluster centroids. A row of the matrix $H$ can be compared to a cluster center. Sorting the values of a cluster center thus corresponds to sorting a row in matrix $H$.

(a) NMF  (b) k-means

Fig. 4: Graphical representation of the feature vectors / cluster centroids

**POS-Tagging** Through POS tagging (Part-Of-Speech-Tagging), it is possible to filter automatically according to certain types of words. When naming clusters, this is very handy, as it can effectively filter out unwanted words. The Stanford POS tagger\(^2\) was used for this work, which in addition to English also supports the German language by using the Stuttgart-Tübingen Tag Set (STTS) \(^2\). The Stuttgart-Tübingen tag set contains over 50 different tags and can, for example, also distinguish between different types of conjunctions.

The following tags were used:

**NN** Normal noun, such as table, apple, travel

**NE** Proper noun, such as Smith, Hamburg

**FM** Foreign language material

Support tickets contain many technical terms, which often come from English, such as domain record or virtual host. In order to exclude these terms as a description for a cluster, in addition to nouns and their own names, they were also filtered according to foreign-language material. All words that do not belong to any of the tags are excluded. For example, POS tagging would remove the words erreichbar and seit from the list server, erreichbar, fehler, managed, seite, problem, ftp, seit, ssh, datenbank.

\(^2\)https://nlp.stanford.edu/software/tagger.shtml
Stemming  We used the Snowball Stemmer for stemform reduction. For words whose word stem occurs more than once, only the original of the first occurrence was retained. The list domain, domains, relocation becomes after stemming domain, relocation.

Manual adjustment  Manual adjustment is the last step in cluster naming. After the word lists have been filtered with the aid of POS tagging and stemming, the most meaningful word, as well as a list of the \( n \) most meaningful words in a file, is stored for each cluster. The entries can be customized by the user.

4  Evaluation

This chapter contains the results of categorization using k-means and NMF. The data set contains only customer requests. This means that only the first message, which the customer writes, is contained. Tickets from employees, as well as automatically generated tickets, are not included. This data set was selected because the categorization of the customer requests was the most useful. For example, if new customer requests are categorized directly, they can be edited by employees who have the most expertise in the category. To include all the tickets form conversation would be counterproductive at this point, since only the first ticket of the customer is available at the time of the categorization of new tickets. The data set contains the latest 50,000 tickets with an average length of 680 characters. The tickets thus cover a period of approximately one year.

Figure 5 shows the results of categorization with k-means and NMF in 30 categories. The categorization was also executed with 50 and 70 categories. Because of the size, the results are not included.

Three different aspects are analyzed for the evaluation, the intersection of categories at k-means and NMF, the distribution of tickets per category, and the development of a category when the number of clusters changes.

4.1 Intersection of categories at k-means and NMF

Both algorithms, k-means and NMF, find similar categories and there is a large intersection between the categories. In categorization with 30 categories, 21 categories are found in the results of both algorithms. In categorization with 50 categories, 33 are equivalent. With 70 categories, there are still 46 of categories equivalent. By increasing the number of categories from 30 to 50, the number of identical categories is decreasing in percentage. When increasing from 50 to 70 categories, there is no longer a large percentage change. With 30 categories, there are 70%, with 50 categories 66%, and with 70 categories 65.7%.

For the categorization of the tickets in 30 categories, the intersection of tickets was examined. For all categories found by both k-means and NMF, it was
checked how many tickets are included in both categories (see fig. 5). Categories that are duplicated under the same or a similar name have been summarized for the calculation of the intersection. For some categories, such as ssl, the intersection is relatively high. Only about 100 tickets from over 2000, which fall into category ssl at k-means, are not in category ssl at NMF. In some categories, however, the number of tickets included varies greatly. In the category password, NMF has more than twice as many tickets as with k-means. However, the category at NMF contains almost all the tickets, which also fall into this category at k-means. With k-means, some of the small categories are even smaller than with NMF.

4.2 Distribution of tickets per category

To better evaluate the distribution of tickets per category, the number of tickets per category is shown in Figure 5. The diagram shows the distribution from the categorization in 30 categories. It is striking that there is a category with a lot of tickets at k-means. When categorized into 30 categories, the largest category contains almost 16,000 tickets. With NMF, the largest category, however, contains only about 5000 tickets, that is only a third of the tickets. A similar pattern can be observed at the categorization with 50 and 70. In 70 categories, the largest category, with nearly 10,000 tickets, contains a fifth of the complete data set. For NMF, however, the largest category is only about 1500 tickets. The difference between k-means and NMF in the largest categories also increases with increased cluster number. Also it is interesting that the name of the largest category in k-means varies depending on the number of clusters. The names of the largest categories are fehler, seite, and account. NMF tends to find categories that are
similar in size. There are no very large categories and no very small categories.

In order to better evaluate whether k-means further reflects the actual structure of the tickets, the large category *fehler* was further analyzed. All tickets assigned by k-means to the category *fehler* have been categorized with NMF. Only 7% of the tickets were assigned with NMF in the same category *fehler*. The largest block with approximately 27% of tickets falls with NMF into the category *seite*. The other tickets split relatively evenly in 26 other categories. One possible theory for the large categories is that with k-means *outliers*, i.e. tickets, which are located at the edge of a cluster, are grouped.

4.3 Development of a category when the number of clusters changes

So far differences between the two algorithms k-means and NMF have been analyzed. In this section we will examine the development of a category at different clusters, but using the same algorithm. The *domain* and *domains* categories were selected as examples. The term *domain* is very general in itself and is experienced in many other areas such as e-mail, SSL certificates or DNS. For this reason the category *domain* is particularly interesting. All 50,000 tickets of the data set were divided into 30, 50 and 70 categories for evaluation with NMF. In 30 categories, a total of 4020 tickets fall into the category *domain(s)*. Out of 4020 tickets, 78% of tickets are again in the *domain(s)* category. The remaining tickets are divided into 22 other categories. The right column shows the categorization with 70 categories. Only 38% of the tickets fall into the category *domain(s)*. Overall, the tickets are now divided into 39 categories. Except for the category *ftp*, all categories from nmf50 were also found by nmf70. That means a larger amount of tickets does not suddenly change into another category, which would be an argument against the quality of the categorization.

5 Advantage of this solution

Compared to a keyword driven approach, categorization with clustering algorithms has a great advantage: Tickets, which do not contain the name of a category, can be assigned to a category. Tickets of this kind are difficult to categorize through simple procedures such as a keyword search, where many/some keywords from predefined lists of keywords for a category need to be present in a text to label it. Ko and Seo used in [3] a keyword driven approach, but would also able to assign a ticket to a category, although the name of the category is not present in the ticket, because they only created training data using the keywords. The difference is that they first had to create categories or keywords, which is not necessary in this approach.

Listing [1.1] shows a ticket which has been assigned by NMF to the category *typo* (as a result of categorization in 30 categories). *typo* stands for the content management system TYPO3, which is used by many Mittwald customers.

Listing 1.1: Example Ticket - Category *typo*
Pfad Sendmail
Hallo! Wie lautet der Pfad zu Sendmail?
Beste Grüße xxxxxx xxx

The ticket itself does not contain the word typo and also no obvious keyword, which suggests a relation to TYPO3. However, Sendmail is a software to send e-mails, which is often used in TYPO3 for this purpose. The path to Sendmail must be entered in TYPO3. Our approach allows this ticket still to be assigned automatically to a meaningful category with the help of algorithms like k-means or NMF.

Other examples are shown in Listings 1.2 and 1.3 where a mail-related and a ssl-related ticket were correctly recognized and assigned to the category mail and ssl, resp.

Listing 1.2: Example Ticket - Category: mail
1 Required TLS in Required Mode
2 Sehr geehrtes Mittwald Support Team
3 Mein Kunde wollte wissen, ob die Mittwald Postfächer als Verschlüsselungsart
4 "TLS im Required Mode" unterstützen. Ich weiss nicht ob es die Starttls
5 Methode ist.
6 Vielen Dank und Gruss xxxxxx xxxxxxx

Listing 1.3: Example Ticket - Category: ssl
1 Let’s encrypt
2 Wie kann ich bei Mittwald eine Let’s encrypt Verschlüsselung nutzen?

The ticket in Listing 1.4 has been assigned to the category php. In this case a keyword driven approach would have failed.

Listing 1.4: Example Ticket - Category: php
1 Execution Time
2 Hallo! Bitte ändern:
3 max_execution_time: 30 Recommended min value 120
4 max_input_vars: 1000 Recommended min value 3000
5 Danke lg xxxxxxx

Assuming manual categorization takes about 4 seconds and there are about 120 customer requests per day, then 8 minutes of time can be saved per day. The automatic categorizing of inventory tickets saves approximately 110 hours.

It should also be noted that the categories found in this domain are quite general and thus give a good overview of the topics that the customers deals with.

6 Conclusion and Future Work

The goal of this work was the unsupervised categorization of support tickets. According to latest research, there is no distinct solution for this. Especially the automatic naming of the calculated categories is not easy to solve. The result of a categorization is strongly dependent on the data set and the cluster algorithm. In addition, the quality of the categorization is difficult to evaluate. This work
has shown that a fully automatic categorization, including the naming of the categories, is feasible and the found categories reflect meaningful subjects.

To improve the quality of the found categories we propose hierarchical clustering of the support tickets. Such a hierarchy could contain more information than a flat clustering. Each category can be divided into other (sub-)categories, providing a more detailed view of the data.

Acknowledgments

We would like to thank Mittwald CM Service GmbH & Co. KG for the cooperation and the provision of the data set.

References

Robust Spectral Clustering for Noisy Data
Modeling Sparse Corruptions Improves Latent Embeddings

Aleksandar Bojchevski, Yves Matkovic, and Stephan Günnemann
Technical University of Munich, Munich, Germany
Department of Informatics and Institute for Advanced Study
{a.bojchevski,matkovic,guennemann}@in.tum.de

Abstract. Spectral clustering is one of the most prominent clustering approaches, with a multitude of successful applications from computer vision to network analysis. Since spectral clustering relies on a similarity graph only, it is applicable in many domains. However, one of the biggest limitations of spectral clustering is its sensitivity to noisy input data.
In this work, we introduce a principle to robustify spectral clustering. We propose a sparse and latent decomposition of the similarity graph used in spectral clustering. Thus, instead of operating on the original graph, the spectral clustering will be performed on the latent representation. In our model, we jointly learn the spectral embedding as well as the latent decomposition – thus, enhancing the clustering performance overall. We provide algorithmic solutions for our model for all three established versions of spectral clustering using different Laplacian matrices. For our solutions we relate to principles such as Eigenvalue perturbation and the multidimensional Knapsack problem. In each case, the complexity of the overall method is linear in the number of edges. Our experimental analysis confirms the significant potential of our approach for robust spectral clustering, with up to 15 percentage points improvement in accuracy on real-world data compared to standard spectral clustering. Moreover, we propose two novel measures – local purity and global separation – which enable us to evaluate the intrinsic quality of an embedding without relying on a specific clustering technique.

References
Abstract. Early machine learning research was strongly interrelated with research on human category learning while later on the focus shifted to the development of algorithms with high performance. Only recently, there is a renewed interest in cognitive aspects of learning. Machine learning approaches might be able to model and explain human category learning while cognitive models might inspire new, more human like, approaches to machine learning. In cognitive science research there exist different theories of category learning, especially, rule-based approaches, prototypes, and exemplar-based theories. To take account of the flexibility of human learning and categorization we propose a human like learning algorithm. In the algorithm we combine incremental decision tree learning, least general generalization, and storing examples for similarity-based categorization. In this paper we present first ideas of this algorithm.

Keywords: (human) supervised category learning, cognitive modeling, incremental decision trees, least general generalizations

1 Introduction

Early machine learning research strongly related to human category learning (cf. Michalski, Carbonell, & Mitchell, 1983). For example, decision tree algorithms were inspired by research on human category learning by Bruner, Goodnow, and Austin (1956). They investigated how humans learned conjunctive or disjunctive categorization rules for a fixed set of artificial stimuli. It could be shown that most humans learned such rules in an incremental way using for example the wholist strategy, that is, they generate an initial rule which is sequentially modified for new examples which do not confirm to the current rule. Based on these findings Hunt, Marin, and Stone (1966) developed the first decision tree algorithms. Subsequently, Unger and Wysotzki (1981) introduced the incremental decision tree algorithm Cal2 as an ideal model for human category learning and Quinlan (1986) developed the well-known ID3.
Starting in the 1990s, machine learning research was focused on the development of new algorithms with high performance and not on cognitive plausibility. Therefore, machine learning and cognitive modeling research separated (Langley, 2016). For example, a main characteristic of human learning is life-long learning, that is, incremental and cumulative learning, while in the field of machine learning, many batch learning algorithms were developed (e.g., ID3, neuronal network approaches, Bayes classifier; Mitchell, 1997). The lack of research in the field of incremental and cumulative and therefore life-long machine learning was addressed and motivated by human learning recently but it did not focus on cognitive plausibility of the algorithms (Thrun, 1998).

Another main characteristic of human learning is the flexible use of different strategies while learning and while using the learned knowledge in categorization tasks. That is, in human category learning there is evidence for rule-based learning, prototype learning, and exemplar-based learning which are combined in hybrid theories of human category learning (Kruschke, 2008). In machine learning, multistrategy learning has been focused in the 1980s and 1990s (Michalski, 1993; Langley, 2016) and Langley (2016) has suggested that the research should continue in this line. Multistrategy learning shares characteristics with ensemble learning. However, while in ensemble learning focus is on combining different models to improve classification performance, in multistrategy learning, focus is on achieving human-level flexibility in application of different knowledge structures in different contexts (Dietterich, 2002).

Besides, Langley (2016) has recommended that researchers should work interdisciplinary because cognitive science and machine learning can mutually profit from each other. Machine learning approaches might be able to model and explain human category learning while cognitive models might inspire new, more human-like, approaches to machine learning. Following these thoughts, we currently are developing a human-inspired learning approach based on Cal 2. Consequently, we focus on learning in supervised classification settings, or in other words in categorization learning from labeled examples.

Therefore, in the following section we describe Cal 2 as the incremental rule-based basis of our algorithm and ID3 which offers an idea for exemplar-based human category learning. Then we introduce least general generalizations (Mitchell, 1977) which inspire a prototypical view of learning. The algorithm combining these ideas is presented and applied to a small example in Section 3. In the last section we discuss further steps like possible extensions of our learning algorithm and the evaluation of the proposed approach.

2 Basic Symbolic Approaches to Category Learning

In this section we first describe relevant decision tree learning algorithms and how they connect to human category learning theories. However, not all aspects of human category learning can be explained with rules as generated with decision trees. Therefore, in the second part of this section we take a closer look on least
general generalizations which could relate to the prototype theories of human category learning.

2.1 Decision Tree Learning, Rules and Exemplars

Algorithm 1 shows the decision tree learner Cal 2 (Unger & Wysotzki, 1981). The algorithm handles examples in an incremental way where the information of previous examples is stored implicitly in the tree structure, but there is no knowledge of previous or later examples given explicitly in each step. The algorithm terminates successfully if all examples are classified correctly. Successful termination is guaranteed for linear separable examples, that is, disjunctive categories. A typical reason for non-disjunctivness is that the given set of features is not sufficient to discriminate the training examples. If examples are not linearly separable, the algorithm fails in Line 9.

The algorithm does not provide a strategy for feature selection. However, the next feature (cf. Line 9) can make use of a predefined selection strategy. In the simplest case the algorithm could choose a feature randomly or use a predefined feature order. The well-known information gain as proposed in ID3 (Quinlan, 1986) is no feasible feature selection criterion for Cal 2 because to calculate the information gain of a specific feature all examples have to be known in advance and therefore the learning is not incremental anymore. However, we currently investigate incremental variants of information gain, for example, the igain measure (Zeller & Schmid, 2016). Calculating an incremental variant of the information gain implies that the already used examples are stored explicitly and not only implicitly in the tree structure. Storing examples explicitly refers to the exemplar-based theories in cognitive psychology (Nosofsky & Palmeri, 1997; Jäkel, Schölkopf, & Wichmann, 2008).

Algorithm 1

Incremental Decision Tree Algorithm Cal 2 (Unger & Wysotzki, 1981)

1: procedure Cal 2(examples)
2: Start with a tree containing only * as node
3: while at least one of the examples changes the tree structure do
4: if current example is classified correct then
5: do nothing
6: if current example is classified as * then
7: replace * with the class of current example
8: if current example is classified wrong then
9: add the next feature in the tree
10: set for the branch with the feature value of current example the class of current example and set * for all other branches

The combination of Cal 2 and igain was used to model human categorization behavior obtained in an experiment with stimuli in form of lamps (Lafond, Lacouture, & Cohen, 2009). The lamps were described by four features with two
discrete feature values each. Features are defined for the base, upright, shade, and top of each lamp and are given as $F_1, \ldots, F_4$. The categorization task in the experiment is to decide whether a specific feature combination belongs to Category $A$ or $B$. The knowledge structure for categorization has to be induced by the humans from nine training examples. Category distribution follows the in psychology often used 5-4 structure (see Table 1; Medin & Schaffer, 1978). The gain of the features helped to explain differences in decision trees when the presentation order of input examples where different for the learner (Zeller & Schmid, 2016). While ordering effects in incremental learning were addressed in early machine learning (cf. Fisher, 1993; Langley, 1995) it only recently comes into focus in the field of cognitive modeling of human category learning (cf. Carvalho & Goldstone, 2015; Mathy & Feldman, 2016).

Table 1. The 5-4 category structure (cf. Medin & Schaffer, 1978) and the input example order for the example in Figure 1.

<table>
<thead>
<tr>
<th>Example</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$F_4$</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$A$</td>
</tr>
<tr>
<td>$e_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$A$</td>
</tr>
<tr>
<td>$e_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$A$</td>
</tr>
<tr>
<td>$e_4$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$A$</td>
</tr>
<tr>
<td>$e_5$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$A$</td>
</tr>
<tr>
<td>$e_6$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$B$</td>
</tr>
<tr>
<td>$e_7$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$B$</td>
</tr>
<tr>
<td>$e_8$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$B$</td>
</tr>
<tr>
<td>$e_9$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$B$</td>
</tr>
</tbody>
</table>

Order: $e_8, e_2, e_5, e_6, e_7, e_1, e_4, e_9, e_3$

2.2 Least General Generalizations and Prototypes

In human category learning there is strong evidence that, especially for natural categories, humans do not learn feature combinations (rules) but prototypes (Rosch & Mervis, 1975). Categories over entities with discrete features are formed by family resemblance, that is, a prototype is given by the most frequent feature value for each feature. It could be shown that humans often search for similarity by (implicit) counting of the occurrence of feature values within a category and compare it with the occurrence of feature values in the other categories.

A similar idea is described as least general generalizations in machine learning where a pattern for a set of examples of a class is generated (cf. Mitchell, 1977). This pattern includes for each feature either a concrete feature value that matches with all examples or a wild card if there exists more than one feature value for this feature in the set of examples. Least general generalizations can be formed for symbolic features, but also for structural features, terms and graphs.
3 An Incremental Decision Tree Algorithm Including Most Specific Patterns and Examples

Algorithm 2 shows our human-inspired decision tree learning approach which realizes incremental learning of rules. Furthermore, to take account of the human flexibility of the categorization process, it integrates storage of generalized patterns and of examples. That is, dependent on the context of a categorization task the learned knowledge structure allows to categorize a new object by applying a rule, by pattern matching, or by similarity to known examples.

Core of our algorithm is Cal2 which can be seen in the procedure INCLUDE (cf. Line 5) where a new example is included to a given tree. To take account of the flexibility of human categorization, the decision tree is enriched by least general generalizations at each decision node and the leaves (cf. most specific pattern, for example, in Line 23). Besides, the examples are stored at the leaves (cf., for example, Line 24). In difference to Cal2 our algorithm considers every input example only once and stores it in the matching current leaf. For this first version of the algorithm which is restricted to disjunctive categories, termination conditions are inherited from Cal2. We have realized this algorithm in Prolog and are currently investigating its ability to model reported findings of human category learning.

The result of the algorithm depends on the category structure, the order of the input examples, and the feature selection criterion. Figure 1 shows the generated tree, with the 5-4 category structure mentioned above and the order of the input examples given in Table 1 when using the igain for feature selection. Left branches show the branches with feature value 1, right branches show the branches with feature value 0. The least general patterns contain the feature values with the following structure: \(< F_1, F_2, F_3, F_4 >\). The ? stands for the wildcard.

4 Future Work

The proposed algorithm is work in early progress and further aspects need to be considered: extension of the learning algorithm and evaluation of the proposed approach.

The current algorithm fails if a new example is categorized erroneously and there is no feature available to extend the tree such that correct categorization becomes possible. This category structure can, for example, be generated by non-disjunctive categories. Learning scenarios involving overlapping categories are often used in psychological studies to demonstrate that humans do not learn rules (Kruschke, 2008). We propose to keep the rule-based structure but to sample examples at the leaf nodes and postpone branching decisions until a certain
Algorithm 2 Incremental Decision Tree Algorithm Including Most Specific Patterns and Examples

1: procedure INCRTMSP(examples)
2:    tree ← empty tree
3:    for each example e ∈ examples do
4:        tree ← INCLUDE(e, tree)
5:    return: tree

5: procedure INCLUDE(e, tree)
6:    new tree ← empty tree
7:    if tree is empty then
8:        new tree ← (most specific pattern of e, label of e, e)
9:    else if tree is leaf with the same class as e then
10:       all examples ← all examples of the leaf and e
11:       new tree ← SAME(all examples)
12:    else if tree is leaf with a different class as e then
13:       all examples ← all examples of the leaf and e
14:       new tree ← DIFFERENT(all examples)
15:    else
16:       branch ← the branch of tree matching with e
17:       node msp ← update msp of root node of branch with e
18:       new branch ← (node msp, INCLUDE(e, branch))
19:       new tree ← substitute branch with new branch in tree
20:    return: new tree

21: procedure SAME(all examples)
22:    branch ← empty tree
23:    msp ← most specific pattern of all examples
24:    branch ← (msp, label of all examples, all examples)
25:    return: branch

26: procedure DIFFERENT(all examples)
27:    branch ← branch all examples by a feature not yet used
28:    for each attribute in branch do
29:        if attribute has no examples then
30:            node ← empty leaf
31:        else if all examples in attribute have the same class then
32:            branched examples ← examples of attribute
33:            node ← SAME(branched examples)
34:        else
35:            branched examples ← examples of attribute
36:            node msp ← most specific pattern of branched examples
37:            node ← (node msp, DIFFERENT(branched examples))
37:        branch ← add node for the attribute of the branch
38:    return: branch
Fig. 1. Generated tree for the 5-4 category structure and the the input example order stated in Table 1. Left branches show the branches with feature value 1, right branches show the branches with feature value 0. The least general generalizations contain the feature values with the following structure: $< F_1, F_2, F_3, F_4 >$. The ? stands for the wildcard.

Amount of evidence is reached. That is, we introduce "delayed branching" that can reduce but not solve the problem of failing.

Delayed branching typically leads to decisions based on stronger evidence, because a larger sample of examples offers a better basis for pattern generalization as well as for similarity-based categorization. Besides, delayed branching might help to reduce overfitting. While in the tree given in Figure 1 most leafs are associated with patterns representing feature vectors and single examples, trees constructed with delayed branching will less often specialize in such a way.

A simple approach to sampling is just to define a threshold for the number of examples needed before a new branch is introduced. This idea is realized in Cal 3 (Unger & Wysotzki, 1981), using a parameter for sample size. We plan to investigate more sophisticated criteria such as prior probabilities (Frermann & Lapata, 2016), or similarity-based coherence of examples (Michalski & Stepp, 1983).

Besides, the least general generalizations does not reflect the prototype approach in detail. That is, prototypes are built by the most frequent feature values, while the least general generalizations represents the feature value that are common among all examples. We plan a variant of the algorithm where at each node the feature values with the highest frequency instead of the least general generalizations is annotated.

Additionally, it might be interesting to make restructuring of the tree and forgetting of examples possible. In human category learning there is evidence that humans completely reject partially learned rules and start with a new rule (Unger
& Wysotzki, 1981). If we assume that wrong information of categories, that is
noise, is seldom we could handle noise by forgetting seldom seen examples and
restructure the tree. This procedure reflects representational shifts (Johansen &
Palmeri, 2002).

Currently, categorization of a new example—while learning and while using
the tree—is strictly guided by the branching in the decision tree. That is, patterns
and exemplars have no influence on the learning and categorization. Hybrid
theories in human category learning take into account that humans are flexible
in using different strategies in different situations (Kruschke, 2008; Nosofsky,
Palmeri, & McKinley, 1994; Rosch, 1983). The factors for using one or another
information in the tree need to be investigated, and incorporated in the current
algorithm.

Since our goal is to develop a human like machine learning algorithm as
well as a machine learning inspired cognitive model of categorization learning,
evaluation of the algorithm has to address performance characteristics as well
as validity as a cognitive model. We plan to take a closer look on efficiency
(time, number of training examples) and accuracy in comparison with other ma-
chine learning approaches, such as (other) decision tree learners, random forests,
and inductive logic programming (Schmid, Zeller, Besold, Tamaddoni-Nezhad,
& Muggleton, 2017).

Furthermore, we want to compare the learning steps as well as the resulting
knowledge structure of our algorithm with human behavior and we plan to pre-
dict human behavior with our algorithm. For this aim we want to select several
learning domains which have been introduced in cognitive science literature. We
are especially interested in domains where it was shown that humans make use
of different strategies for learning and categorization. Among these domains is
the lamp domain introduced above but also artificial domains of letter strings,
or natural categories such as fruit (Rosch & Mervis, 1975).

References

Carvalho, P. F., & Goldstone, R. L. (2015). What you learn is more than
what you see: What can sequencing effects tell us about inductive category
Dietterich, T. G. (2002). Ensemble learning. The handbook of brain theory and
in Incremental Learning: AAAI Spring Symposium at Stanford University
Frermann, L., & Lapata, M. (2016). Incremental Bayesian category learning


Interpretable Matrix Factorization with Stochasticity Constrained Nonnegative DEDICOM

Rafet Sifa\textsuperscript{1,2}, Cesar Ojeda\textsuperscript{1}, Kostadin Cvejoski\textsuperscript{1}, and Christian Bauckhage\textsuperscript{1,2}

\textsuperscript{1} Fraunhofer IAIS, Sankt Augustin, Germany
\textsuperscript{2} University of Bonn, Bonn, Germany
www.multimedia-pattern-recognition.info
\{name.surname\}@iais.fraunhofer.de

Abstract. Decomposition into Directed Components (DEDICOM) is a special matrix factorization technique to factorize a given asymmetric similarity matrix into a combination of a loading matrix describing the latent structures in the data and an asymmetric affinity matrix encoding the relationships between the found latent structures. Finding DEDICOM factors can be cast as a matrix norm minimization problem that requires alternating least square updates to find appropriate factors. Yet, due to the way DEDICOM reconstructs the data, unconstrained factors might yield results that are difficult to interpret. In this paper we derive a projection-free gradient descent based alternating least squares algorithm to calculate constrained DEDICOM factors. Our algorithm constrains the loading matrix to be column-stochastic and the affinity matrix to be nonnegative for more interpretable low rank representations. Additionally, unlike most of the available approximate solutions for finding the loading matrix, our approach takes the entire occurrences of the loading matrix into account to assure convergence. We evaluate our algorithm on a behavioral dataset containing pairwise asymmetric associations between variety of game titles from an online platform.

Keywords: Unsupervised Learning, Matrix Factorization, Constrained Optimization, Behavior Analysis

1 Introduction

Matrix and tensor factorization methods have been widely used in data science applications for mostly understanding hidden patterns in the datasets and learning representations for variety of prediction tasks and recommender systems [1, 5, 10, 11, 13, 16, 17]. Decomposition into Directed Components (DEDICOM) [8] is a matrix and tensor factorization technique and a compact way of finding low rank representations from asymmetric similarity data. The method has found applications in various data science problems including analysis of temporal graphs [1], first order customer migration [15], natural language processing [4], spatio-temporal representation learning [2, 16] and link prediction in
Fig. 1: Illustration of two-way DEDICOM to partition a given similarity matrix $S \in \mathbb{R}^{n \times n}$ encoding pair-wise asymmetric relations among $n$ objects into a combination of a loading matrix $A \in \mathbb{R}^{n \times k}$ and a square affinity matrix $R \in \mathbb{R}^{k \times k}$. In this representation $A$ contains the latent factors where the columns describe the hidden structures in $S$ and the relations among these structures are encoded by the asymmetric affinity matrix $R$. Variety of constrained versions of DEDICOM has been previously studied so as to improve the interpretability and the performance of finding appropriate factors.

Consider the three factor approximations of the similarity values as in (1) and (2), the interpretation (especially with respect to scale) of the hidden structures from given factor matrices $A$ and $R$ might yield misleading results with the presence of negative values [15]. That is, the interpretation of the results is limited to only consider nonnegative affinities in $R$ or the positive or negative loadings of the corresponding points in (2). Additionally, when analyzing nonnegative data matrices (such as ones containing probabilities, counts and etc.), it is usually beneficial to consider nonnegative factor matrices that can be considered as condensed (or compressed) representations that can be used for informed decision making and representation learning [1,8,15,16].
In this work we address the issue of interpretability of the DEDICOM factors by introducing a converging algorithm as an alternative to the previously proposed methods in [1,13,15,16]. Our method constrains the loading matrix $A$ to contain column stochastic vectors and (similar to [15,16]) the affinity matrix $R$ to be nonnegative. Formally this amounts to factorize a given asymmetric similarity matrix $S$ as in (1) by enforcing

$$r_{ij} \geq 0 \forall \{i,j\}$$

and

$$a_{cb} \geq 0 \land \sum_{q=1}^{n} a_{q}b = 1 \forall \{c,b\}.$$ (4)

Compared to the additive-scaling based representation introduced in [15,16], constraining the columns of $A$ to contain probabilities has the direct advantage of interpreting each element of $A$ as the representativeness value of the particular loading it represents. That is, the value of $a_{ib}$ represents how much the $i$th element in the dataset contributes to the $b$th latent factor. This is indeed parallel to the idea of Archetypal Analysis [3,6,17], where the mixture matrices respectively determine how much a data point and archetypes contribute to respectively constructing the archetypes and reconstructing each data point using the archetypes.

In the following we will first describe the general alternating least squares optimization framework and give an overview of the algorithms to find appropriate constrained and unconstrained DEDICOM factors. After that we will introduce our algorithm by first studying our problem setting and deriving the algorithm step-by-step. This will be followed by a case study covering a real world application where we will analyze game-ownership patterns from data containing asymmetric associations between games using the factors we extract by running our algorithm. Finally, we will conclude our paper with a summary and some directions for future work.

2 Algorithms for Finding DEDICOM Factors

Finding appropriate DEDICOM factors for matrix decomposition can be cast as a matrix norm minimization problem with the objective

$$E(A, R) = \| S - ARA^T \|^2,$$ (5)

which is minimized with respect to $A$ and $R$. Non-convex minimization problems of this kind usually follow an iterative alternating least squares (ALS) procedure where at each iteration we optimize over a selected factor treating the other factors fixed. It is important to note that, the minimized objective function in (5) is convex in $R$ for fixed $A$ whereas not convex in $A$ for fixed $R$, which leads to optimal and sub-optimal approximate solutions when respectively updating $R$ and $A$. 
Starting with defining update rules for the affinity matrix $R$, we note that with fixed $A$, minimization of (5) is a matrix regression problem [1,13,15] with global optimum solution

$$R \leftarrow A^\dagger SA^T,$$  \hspace{1cm} (6)

where $A^\dagger$ indicates Moore-Penrose pseudoinverse of $A$. Furthermore, if $A$ is constrained to be column orthogonal (i.e. $A^T A = I_k$, where $I_k$ is the $k \times k$ identity matrix), the update for $R$ simplifies to $R \leftarrow A^T SA$ [15]. Here the updates are not constrained to fall into a particular class of domain and might result in with affinity matrices containing negative elements. Constraining $R$ when minimizing (5) to contain only nonnegative values can be cast as a nonnegative least squares problem by transforming the arguments of (5) as

$$E(R) = \left\| \text{vec}(S) - \text{vec}(ARA^T) \right\|^2$$  \hspace{1cm} (7)

where the vec operator vectorizes (or flattens) a matrix $B \in \mathbb{R}^{m \times n}$ as

$$\text{vec}(B) = [b_{11}, \ldots, b_{m1}, \ldots, b_{1n}, \ldots, b_{mn}]^T.$$

(8)

It is worth mentioning that since vectorizing (5) as shown in (7) does not affect the value of the norm [15], we can make use of the property of vectorization to represent a vectorization of multiplications of matrices as matrix vector multiplication to rewrite (7) as

$$E(R) = \left\| \text{vec}(S) - (A \otimes A) \text{vec}(R) \right\|^2,$$  \hspace{1cm} (9)

where $\otimes$ represents the Kronecker product. As noted in [15], the expression in (9) can indeed be mapped to constrained linear regression problem [12] with a converging result (through an active set algorithm) to define an update as

$$R \leftarrow \arg\min_R \left\| \text{vec}(S) - (A \otimes A) \text{vec}(R) \right\|^2 \land r_{ij} \geq 0 \ \forall \ i,j.$$

(10)

Having identified alternating least squares updates for constrained and unconstrained affinity matrix $R$, we now turn our attention to define updates for the loading matrix $A$. To this end, there have been two different directions followed previously: approximating the minimization of (5) by treating a particular factor with $A$ constant [1,13] and directly minimizing (5) by means of gradient based approaches. To start with, former method considers stacking matrix $S$ and matrix $S^T$ as

$$[S \ S^T] = [ARA^T \ AR^T A^T] = A[R A^T \ R^T A],$$

(11)

and solves for $A$ keeping $A^T$ fixed [1,13] to come up with an update for $A$ as

$$A \leftarrow \left([SAR^T + S^T AR] (R A R^T A^T + R^T A R)^{-1}\right).$$

(12)
The latter method’s ALS update for the loading matrix $A$ is based on directly minimizing (5) by considering first order gradient based optimization [15, 16], which in each update moves the current solution for $A$ in the opposite direction of gradient of (5). To derive the gradient matrix we start by defining (5) in terms of the trace operator as

$$E(A, R) = \text{tr} \left[ S^T S - S^T ARA^T - AR^T A^T S + AR^T A^T ARA^T \right]$$
$$= \text{tr} \left[ S^T S - 2S^T ARA^T + AR^T A^T ARA^T \right]$$
$$= \text{tr} \left[ S^T S \right] - 2 \text{tr} \left[ S^T ARA^T \right] + \text{tr} \left[ AR^T A^T ARA^T \right].$$

(13)

Since traces are linear mappings and the term $\text{tr} \left[ S^T S \right]$ does not depend on $A$, minimizing $E(A, R)$ in (13) with respect to $A$ is equivalent to minimizing $E(A)$ which we define as

$$E(A) = E_1(A) + E_2(A)$$

given that

$$E_1(A) = -2 \text{tr} \left[ S^T ARA^T \right]$$

(15)

and

$$E_2(A) = \text{tr} \left[ AR^T A^T ARA^T \right].$$

(16)

Considering the orthogonality constraint on $A$, the term in (16) becomes independent of $A$. That is, since traces are invariant under cyclic permutation we can reformulate (16) as

$$E_2(A) = \text{tr} \left[ AR^T A^T ARA^T \right] = \text{tr} \left[ R^T A^T ARA^T A \right] = \text{tr} \left[ R^T R \right],$$

(17)

which allows for only considering the minimization of the error term in (15) [15, 16]. Consequently, we can define a gradient descent based update by considering the gradient matrix

$$\frac{\partial E_1(A)}{\partial A} = -2 \left( S^T AR + SAR^T \right),$$

(18)

which with a learning rate $\eta_A$ allows us to define an update for $A$ as

$$A \leftarrow A + 2\eta_A \left( S^T AR + SAR^T \right).$$

(19)

As a final step, in order to assure the orthogonality constraint on $A$ we project the updated $A$ by considering its $QR$ decomposition [15, 16] $A = QT$, where $Q \in \mathbb{R}^{n \times k}$ is orthogonal and $T \in \mathbb{R}^{k \times k}$ is invertible upper triangular, and following the update $A \leftarrow Q$. This concludes our overview of the methods to come up with appropriate factors. For more detailed information about the alternating least squares solutions for DEDICOM factorization for matrices and tensors we refer the reader to [1, 13, 15, 16].
3 Stochasticity Constrained Nonnegative DEDICOM

In this section we will present a new ALS algorithm to particularly consider DEDICOM factorization with column stochastic loadings and nonnegative affinities for interpretability of the resulting factors. To begin with, as noted in [15,16], we can assure nonnegative affinities $R$ by solving the nonnegative least squares problem introduced in [12] as in (10).

Next considering the theoretical properties of constraining the columns of matrix $A$ to (4), we note that each column resides in the standard simplex $\Delta^{n-1}$, which is the convex hull of the standard basis vectors of $\mathbb{R}^n$. Formally, given that $\delta_{ij}$ represents the Kronecker delta and $V = \{v_1, v_2, \ldots, v_n|v_i = [\delta_{i1}, \ldots, \delta_{in}]^T\}$ is the set of the standard basis vectors of $\mathbb{R}^n$, the standard simplex $\Delta^{n-1}$ is defined as the compact set

$$\Delta^{n-1} = \left\{ \sum_{i=1}^{n} \alpha_i v_i | \sum_{i=1}^{n} \alpha_i = 1 \wedge \alpha_i \geq 0 \forall i \in [1, \ldots, n] \right\}. \quad (20)$$

This indicates that, finding an appropriate column stochastic matrix $A$ minimizing the convex objective function in (5) can be reduced down to a constrained optimization problem in the convex compact simplex $\Delta^{n-1}$.

Constrained optimization problems of this kind can be easily tackled using the Frank-Wolfe algorithm [7,9], which is also known as the conditional gradient method [9]. The main idea behind the Frank-Wolfe algorithm is to minimize differentiable convex functions over their domains forming compact convex sets using iterative first-order Taylor approximations until achieving convergence. Formally given a differentiable convex function $f : S \rightarrow \mathbb{R}$ over a compact convex set $S$, the Frank-Wolfe algorithm aims to solve

$$\min_x f(x) \quad (21)$$

for $x \in S$, by iteratively solving for

$$s_t = \min_{s \in S} s^T \nabla f(x_t), \quad (22)$$

where $\nabla f(x_t)$ is the gradient of the optimized function $f$ evaluated at the current solution $x_t$. Following that, at each iteration $t$, the algorithm estimates the new solution by evaluating

$$x_{t+1} = x_t + \alpha_t(s_t - x_t), \quad (23)$$

where $\alpha_t \in (0, \ldots, 1]$ is the learning rate that is usually selected by performing a line search on (23) or set to be monotonically decreasing as a function of $t$ [7,9]. Additionally, one particular advantage of this optimization method is that it allows for $\epsilon$-approximation for a given maximum number of iterations $t_{\text{max}}$ [7,9]. Considering our special case, since the set of vertices of a standard simplex is equivalent to the set of standard basis $V$ in $\mathbb{R}^n$ [3], at each iteration Frank-Wolfe
algorithm moves the current solution into the direction of one of the standard basis until achieving convergence.

Following that, since the stochasticity constraint does not allow for a simplification as in (17), we require the full derivation of the gradient matrix \( \frac{\partial E(A)}{\partial A} \) from (13) so as to adapt the Frank-Wolfe algorithm to find appropriate stochastic columns for \( A \). To this end we start by considering \( E_2(A) \) from (16) in terms of a scalar summation as

\[
\text{tr} \left[ A^T A^T A R A^T \right] = \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz} \tag{24}
\]

and consider its partial derivative with respect to an arbitrary element \( a_{ij} \) of \( A \) that can be written due to linearity of differentiation as

\[
\frac{\partial E_2(A)}{\partial a_{ij}} = \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z \frac{\partial}{\partial a_{ij}} (a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz}) \tag{25}
\]

The expression in (25) can be further simplified by the product rule expansion

\[
\frac{\partial E_2(A)}{\partial a_{ij}} = \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z \frac{\partial}{\partial a_{ij}} (a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz}) + \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z \frac{\partial}{\partial a_{ij}} (a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz}) + \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z \frac{\partial}{\partial a_{ij}} (a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz}) + \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z \frac{\partial}{\partial a_{ij}} (a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz}) \tag{26}
\]

and evaluating the derivatives results in

\[
\frac{\partial E_2(A)}{\partial a_{ij}} = \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z r_{uw} a_{xw} a_{xy} r_{yz} a_{uz} + \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz} + \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z a_{uv} r_{uw} a_{xw} a_{xy} r_{yz} a_{uz} + \sum_u \sum_v \sum_w \sum_x \sum_y \sum_z r_{yz} a_{uv} r_{uw} a_{xw} a_{xy}, \tag{27}
\]

Finally, reformulating the four summations in (27) in terms of matrix multiplications as

\[
\frac{\partial E_2(A)}{\partial a_{ij}} = 2 \left[ A R^T A R \right]_{ij} + 2 \left[ A R A^T A R^T \right]_{ij} \tag{28}
\]
Randomly initialize valid $A$ and $R$

Let $V = \{v_1, v_2, \ldots, v_n | v_i = [\delta_{i1}, \ldots, \delta_{in}]^T\}$

while Stopping condition is not satisfied do
  //Update $A$ with Frank-Wolfe procedure
  while $t \neq t_{\text{max}}$ and updates of $A$ are not small do
    //Define the gradient matrix
    $G = \frac{\partial E(A)}{\partial A} = 2 \left( AR^T A^T A R + A R^T A R^T - S^T A R - S A R^T \right)$
    //Define the monotonically decreasing learning rate
    $\alpha \leftarrow 2/(t + 2)$
    for $b \in \{1, \ldots, k\}$ do
      //Find the minimizer simplex vertex $i$
      $i = \text{argmin}_l g_{bl}$
      //Update columns $a_b$ of $A$
      $a_b \leftarrow a_b + \alpha(v_i - a_b)$
      //Update the iterator
      $t \leftarrow t + 1$
  //Update $R$ by solving the nonnegative least squares problem
  $R \leftarrow \text{argmin}_R \left\| \text{vec}(S) - \left( A \otimes A \right) \text{vec}(R) \right\|^2 \text{ w.r.t } r_{ij} \geq 0 \forall i, j$

 Alg. 1: A Frank-Wolfe based projection-free ALS algorithm to find Seminonnegative DEDICOM factors with column stochastic loading matrix $A$ and nonnegative affinity matrix $R$. At each iteration of the algorithm, we alternate between finding an optimal column stochastic $A$ keeping $R$ fixed and finding a nonnegative matrix $R$ keeping $A$ fixed. Note that the learning rate here is set to decrease monotonically.

allows us to define the gradient matrix $\frac{\partial E_2(A)}{\partial A}$ as

$$\frac{\partial E_2(A)}{\partial A} = 2 \left( AR^T A^T A R + A R^T A R^T \right).$$

(29)

Combining the results from (18) and (29) the full gradient matrix of the minimized error norm for $A$ is calculated as follows

$$\frac{\partial E(A)}{\partial A} = 2 \left( AR^T A^T A R + A R^T A R^T - S^T A R - S A R^T \right),$$

(30)

which allows us to define a column-wise Frank-Wolfe algorithm to come up with optimal column stochastic loading matrix $A$ minimizing (5). We list the essential steps of our adaptation of the projection free Frank-Wolfe algorithm in Alg. 1. In a nutshell, our algorithm starts by randomly initializing (valid) matrices $A$ and $R$, continues with alternating between the Frank-Wolfe optimization updates to come up with optimal column stochastic $A$ and alternating least squares
Fig. 2: Illustration of the evolution of the Residual Sum of Squares (RSS) for factorizing our empirical conditional probability matrix indicating asymmetric relations between games using DEDICOM with \( k = 3 \). Our provably converging algorithm monotonically decreases the RSS at each iteration.

updates for nonnegative \( R \) until a predefined stopping condition is met. The stopping conditions are usually selected based on reaching a maximum number of alternating least squares updates (not to be mixed with the maximum iteration count \( t_{max} \) of the Frank-Wolfe updates for columns of \( A \) in Alg. 1), having minor subsequent changes in the minimized objective (for our case the matrix norm in (5)) or combining both of these conditions.

4 A Business Intelligence Application: Analyzing Asymmetric Game Ownership Associations in an Online Gaming Platform

In order to evaluate our algorithm, we consider a use-case from game analytics [14,16], where we analyze the asymmetric relationships among various types of games. To this end we consider a dataset from an online gaming platform called Steam, which hosts thousands of games of various genres to a large player-base with its size ranging from 9 to 15 million\(^3\) (daily active) players. We used the dataset from [14] which contains game ownership information of more than six million users about 3007 titles.

Representing the ownership information as a bipartite matrix \( Y \in \{0, 1\}^{m \times n} \) indicating ownership of information of \( m \) players for \( n \) games, we construct the asymmetric association among games in terms of their pairwise empirical conditional probabilities. To this end, we define the directional similarity from game \( i \) to game \( j \) as

\[
s_{ij} = \frac{|\{c \mid y_{ci} \neq 0 \forall c \} \cap \{b \mid y_{bj} \neq 0 \forall b\}|}{|\{c \mid y_{ci} \neq 0 \forall c\}|},
\]

\(^3\) [http://store.steampowered.com/stats/](http://store.steampowered.com/stats/)
where \(|·|\) indicates set cardinality and \(y_{pq}\) represents if the \(p\)th player owns the \(q\)th game. It is important to note that (31) is inherently asymmetric and a special case of the so-called Tversky index [18] with \textit{prototype} and \textit{variant} weights of respectively 1 and 0 (or vice versa). Since the number of computations required to obtain the similarity matrix \(S\) scales to \(O(mn^2)\), we parallelized the procedure of computing the similarity values in (31) by utilizing a hybrid parallelization technique that simultaneously exploits both distributed and shared memory architectures in order to speedup the computation time.

Having obtained the asymmetric similarity matrix with the empirical conditional probability values in \(S\), we factorize it using our algorithm and present the results with \(k = 3\). To explicitly ignore modeling the self-loops, at each alternating least squares iteration, we replaced the diagonal of \(S\) with the diagonal of its current reconstruction [1]. In Fig. 2 we show the residual sum of squares, which decreases monotonically and converges to a minimum after 30 iterations. Analyzing the resulting factors, we observe that the resulting two columns (or modes) \(a_1\) and \(a_2\) of \(A\) are sparse whereas the last column was dense in terms of non-zero probability values, where the most dominant games vary from column to column. In Tbl. 1 we present the games with high loadings in their corresponding column and observe that mostly the free-to-play flagship game Team Fortress 2 (TF 2), followed by the FPS games CS: Source and Red Orchestra: Ostfront 41-45, contribute to \(a_1\). On the other hand, the indie-platformer games Wooden Sen’SéY (WSS), Ethan (Meteor Hunter) and Oozi: Earth Adventure contribute mostly to \(a_2\). Finally, the mostly dense \(a_3\) has very high loadings for all of the flagship games of the analyzed platform and other AAA games including Left 4 Dead 2, Dota 2, Skyrim, Half-life 2 (HL 2) and Garry’s Mod.

Analyzing the resulting row-normalized affinity matrix that encodes the asymmetric relations between the modes from Fig. 3, the first rows indicates tendencies of TF 2 players more to the indie-platformer mode than to the AAA games because of the fact that TF 2 is mostly a singular game that people primarily prefer in the Steam platform [14]. On the other hand, inline with the results from [14] the opposite directional associations, namely, from the flagships to the TF 2 are high which is related to the fact that majority of the players playing one or more of the flagship games also prefer TF 2 [14]. Finally analyzing the second mode, similar to the results in [15] the self association is the highest association we observe (indicating players preferring mostly to stay in the same genre), this is followed by high associations to the TF 2 and the mode related to the flagship and AAA games.

<table>
<thead>
<tr>
<th>Loadings</th>
<th>Identifier</th>
<th>Dominant Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)</td>
<td>TF 2/FPS</td>
<td>TF 2, CS: Source and Red Orchestra: Ostfront 41-45</td>
</tr>
<tr>
<td>(a_2)</td>
<td>Indie/Platformer</td>
<td>WSS, Ethan and Oozi: Earth Adventure</td>
</tr>
<tr>
<td>(a_3)</td>
<td>Flagships/AAA</td>
<td>Left 4 Dead 2, Dota 2, Skyrim, HL 2, Garry’s Mod</td>
</tr>
</tbody>
</table>

Table 1: Selection of games with high loading values for the analyzed dataset.
Fig. 3: Results of factorizing the empirical conditional probability graph created for the pair-wise game ownership information of more than six million players of an online gaming platform. The normalized nonnegative affinity matrix $R$ shows relationships between groups automatically determined by the DEDICOM algorithm.

It is worth mentioning that, parallel to the results in [1], by increasing the number of modes $k$ for the loading matrix $A$, we observed more detailed partitions regarding the modeled patterns. Running our algorithm with $k = 2$, we observed a higher reconstruction error and that the indie-platformer mode $a_2$ remains to be one of the factors whereas the games contributing to $a_3$ have merged with ones contributing to mode $a_1$. On the other hand, considering the resulting factors when we ran our algorithm with $k = 4$, we obtained a slight improvement regarding the reconstruction error and observed that the modes $a_1$ and $a_2$ remained the same, however, mode $a_3$ has split into two different modes where both contain the same games as in $a_3$ and the new mode contains additional indie and plafomer games (reducing the probability on the games of $a_3$) such as Finding Teddy, Gentlemen, Gravi and Face Noir.

5 Conclusion and Future Work

In this work we studied the DEDICOM model to factorize asymmetric similarity matrices into a combination of a low rank loading matrix and an affinity matrix. We gave an overall overview about the theoretical details and algorithms to come up with appropriate factors. In essence, the affinity matrix matrix $R$ indicates different directional structures that are determined by the columns of the loading matrix $A$. Having defined our objective function to be the matrix norm of the difference between the factorized asymmetric similarity matrix and its DEDICOM reconstruction, we derived the gradient matrix for this function with respect to the loading matrix $A$ and presented a variant of the Frank-Wolfe algorithm to obtain interpretable column stochastic loadings and nonnegative affinities. Running our algorithm on a dataset containing asymmetric pairwise
relationships between more than 3000 games, we found interesting patterns indicating directional preferences among games.

Our future work involves analyzing the performance of our model for different tasks including link prediction and representation learning [13, 16]. Considering the tensor extensions in [13, 16], our future work also involves extending the proposed model to tensors factorizations which can allow us to analyze asymmetric similarity matrices from different sources. In the context of business intelligence and game analytics, this might allow us to analyze and compare, for instance, preferences of different countries or player groups.

References

PRIMPing Boolean Matrix Factorization by Proximal Alternating Linearized Minimization

Sibylle Hess, Nico Piatkowski, and Katharina Morik
TU Dortmund, Computer Science, LS 8, 44221 Dortmund, Germany

Abstract. We propose a novel Boolean matrix factorization algorithm, based on recent results from optimization theory. We demonstrate the superior robustness of the new approach in the presence of several kinds of noise and the interpretability on synthetic and real-world data.

Keywords: Boolean Matrix Factorization, Minimum Description Length, Proximal Minimization, Nonconvex-Nonsmooth Minimization

Given the task to explore binary data, Boolean Matrix Factorization (BMF) is a method of choice. BMF yields a simultaneous clustering of rows and columns of the data matrix into binary, thus interpretable, cluster representatives. Furthermore, state-of-the art methods automatically estimate the number of prevalent clusters through the application of the minimum description length principle [2].

Unfortunately, existing algorithms are greedy and rely on heuristics to solve the NP-hard problem of BMF. We propose with the procedure PRIMP to apply the optimization scheme PALM to a real-valued relaxation of the objective. PALM enables the minimization of the generally nonconvex description length and the nonsmooth penalization of non-binary values under convergence guarantees. A rounding procedure rounds the result to binary values and decides over the number of returned clusters. For more information, we refer to [1].

Acknowledgments SFB 876, projects A1 and C1.

References
KPCA Embeddings: an Unsupervised Approach to Learn Vector Representations of Finite Domain Sequences
A Use Case for Words and DNA Sequences

Eduardo Brito, Rafet Sifa, and Christian Bauckhage
Fraunhofer IAIS
Schloss Birlinghoven, Sankt Augustin, Germany
https://multimedia-pattern-recognition.info

Abstract. Most of the well-known word embeddings from the last few years rely on a predefined vocabulary so that out-of-vocabulary words are usually skipped when they need to be processed. This may cause a significant quality drop in document representations that are built upon them. Additionally, most of these models do not incorporate information about the morphology of the words within the word vectors or if they do, they require labeled data. We propose an unsupervised method to generate continuous vector representations that can be applied to any sequence of finite domain (such as text or DNA sequences) by means of kernel principal component analysis (KPCA). We also show that, apart from their potential value as a preprocessing step within a more complex natural language processing system, our KPCA embeddings also can capture valuable linguistic information without any supervision, in particular word morphology of German verbs. When they are applied to DNA sequences, they also encode enough information to detect splice junctions.

1 Introduction

Machine learning approaches for natural language processing (NLP) generally demand a numeric vector representation for words. We can distinguish any two different words from a fixed vocabulary by assigning them a one-hot vector, where all entries of the vector are zero-valued but in a single position that identifies the word. This is a very sparse representation that encodes no information about the words but their position in the vocabulary.

A more information-rich alternative to one-hot vectors are the so-called word embeddings. They are distributed vector representations, which are dense, low-dimensional, real-valued and can capture latent features of the word [13]. Based on the distributional hypothesis [5] (words that appear in similar contexts have similar meanings), they exploit word co-occurrence so that similar words are
mapped close to each other in the word vector space. Part of the success of the word embeddings is due to their efficient shallow neural network architectures such as the continuous skip-gram model and the continuous bag of words model [9], widely popularized after the release of word2vec. Word embeddings also inspired research in other areas different from NLP to learn vector representations such as node representations from a graph [4].

Although syntax and semantics can be encoded with word2vec embeddings, they do not incorporate morphological information about the word. As a consequence, morphologically similar words may not be nearby in the word vector space. Some approaches make use of existing linguistic resources so that the word embeddings capture not only contextual information but also morphological information[3, 7]. Due to their dependence on language-specific resources, they will not work for languages whose available linguistic resources are scarce.

The subword-based models such as FastText [1, 6] learn indirectly morphology by learning not only word vectors but also n-gram vectors. This enables not only complete unsupervised learning but also the possibility of inferring out-of-vocabulary (OOV) words, which is an important issue for all other approaches mentioned, notably when noisy informal text needs to be processed. Despite these advantages of subword-based models, some morphologically rich languages (where these models are supposed to perform specially well) may contain very long words. This leads to a dramatic increase of the number of necessary n-gram representations, increasing time and space complexity as well.

We propose an alternative unsupervised method by means of kernel principal component analysis (KPCA) that encodes morphology while learning word representations and that generate new vectors for OOV words after training. In addition, our approach is general enough to learn vector representations for any sequence whose elements belong to a fixed predefined finite set. In particular, we test our KPCA embeddings in two different tasks: classifying the verb category for German verbs and detecting splice junctions in DNA sequences.

2 Approach

KPCA indirectly maps vectors to a feature space (of higher dimension) in order to obtain the principal components from that space [12]. No explicit calculation in the feature space is required since we only need to be able to compute the inner product in the feature space and this can be achieved by using kernel functions. We can exploit the freedom to select any inner product of our choice so that we can also perform KPCA to non-numeric entities. Formally, given a zero-mean column data matrix \(X = [x_1, x_2, \ldots, x_n]\) containing \(n\) \(m\)-dimensional data points, principal component analysis (PCA) deals with representing the data through principal components maximizing the variance in \(X\) by solving

\[
Cv = \lambda v, \tag{1}
\]
where $C = \frac{1}{n}XX^T$ is the covariance matrix, $\lambda$ an arbitrary eigenvalue and $v$ its corresponding eigenvector. Considering (1), we can represent every eigenvector as a linear combination of the data points the data matrix $X$ as

$$\frac{1}{n\lambda} XX^T v = X\beta = v,$$

where $\beta \in R^m$. Substituting (2) in (1) we obtain

$$\frac{1}{n} XX^T X\beta = \lambda X\beta,$$

which upon data projection can be represented as

$$\frac{1}{n} X^T XX^T X\beta = \lambda X^T X\beta.$$  

Replacing the occurrences of the gram matrix $XX^T$ by a selected kernel matrix $K$ as

$$\frac{1}{n} KK\beta = \gamma K\beta, \quad (5)$$

and eliminating $K$ as

$$\frac{1}{n} K\beta = \gamma \beta, \quad (6)$$

we result in a kernelized representation of the conventional PCA.

For the particular case of words and DNA sequences, the inner product can be any string similarity. For our experiments, we adapt the Sørensen-Dice coefficient by considering not only bigrams, but n-grams of any length in general. Let $V$ a vocabulary of words, $G_n(w)$ the n-grams of a word $w \in V$. We define our similarity function $s$ of two words $x,y \in V$ as follows:

$$s(x,y) = \sum_{n \in N^+} \alpha_n \frac{2|G_n(x) \cap G_n(y)|}{|G_n(x) \cup G_n(y)|} \cdot \sum_n \alpha_n = 1 \quad (7)$$

where $\alpha_n$ determines the weight of the Sørensen-Dice coefficient term for each n-gram length. We compute a similarity matrix $S$ by applying this similarity function $s$ to all word pairs of our vocabulary $V$:

$$S_{ij} = s(w_i, w_j) \quad \forall w_i, w_j \in V. \quad (8)$$

Then, we calculate the kernel matrix $K$ by applying a non-linear kernel function (for instance RBF kernel or polynomial kernel) to the similarity matrix $S$. After computing the eigenvectors and eigenvalues of the resulting matrix $K$, we construct our projection matrix $P$ by selecting $d$ eigenvectors $v_1, \ldots, v_d$ and dividing them by their respective eigenvalues $\lambda_1, \ldots, \lambda_d$:

$$P = \begin{bmatrix} v_1 & \cdots & v_d \end{bmatrix} \begin{bmatrix} 1/\lambda_1 & & \\ & \ddots & \\ & & 1/\lambda_d \end{bmatrix}. \quad (9)$$

We can now generate a KPCA embedding for any word $w_t$. We only need to compute the similarity function of the word against all the words processed from
Fig. 1: Visualization of KPCA embeddings generated from a reduced German vocabulary with a RBF kernel, $\sigma = 0.7$ and two principal components. Each color represents a different verb tense. Black dots refer to German prefixes. Best seen in color.

the vocabulary $\mathcal{V}$ and apply the kernel function $k$ to the resulting vector. This results in a kernelized distance vector $r_t$. The product of $r_t$ with the projection matrix $P$ constitutes the $d$-dimensional KPCA embedding $u_t$ of the word $w_t$.

$$r_t = k(s(w_t, \mathcal{V})) \quad \text{and} \quad u_t = P^\top r_t. \tag{10}$$

It is important to note that, this approach can be generalized to encode any other non-numeric entity as long as we can define an equivalent similarity function between each pair of entities.

3 Experiments

We show how our KPCA embeddings can be used for different kinds of sequential data. In particular, we apply our approach to represent words to classify different types of verb forms and to represent DNA sequences to recognize splice junctions.
3.1 Verb classification

We test how our KPCA embeddings can encode morphology with a fine-grain POS tagging task. We restrict our vocabulary to the tokens tagged as verbs from the TIGER treebank [2]. This simplifies the problem to classify the correct morphological tag (consisting of grammatical person, number, tense and mode when they apply) of a German verb only from its KPCA embedding.

First, we extract all tokens tagged as verb (corresponding to the TIGER tags VVFIN, VAFIN, VMFIN, VVIMP, VAIMP, VVINF, VVIZU, VAINF, VMINF, VVPP, VMPP, VAPP) and remove all duplicates. This leads to 13370 unique verbs with 31 different morphological tags, whose distribution is showed in figure 2. Then, we apply the approach described in section 2. We build a training set consisting of 80% of the verbs and a test set with the remaining 20%. We consider only bigrams and trigrams to compute the similarity function for each pair of verbs by selecting five different weight distributions (different values for $\alpha_2$ and $\alpha_3$ in equation 7). We also incorporate an additional character at the beginning and at the end of each verb when producing the n-grams. After running KPCA on the training set, we infer vector representations of the verbs from the test
set. By using the KPCA embeddings as features and the morphological tag as label, we train k-nearest neighbors classifiers to predict the morphological tag of a word from only the KPCA embedding. As baseline representations, we also learn a word2vec [10, 9] model for each different vector size. These word vectors were learned applying the default hyperparameter values.

From Table 1 we can observe that a mean accuracy above 77% can be achieved by classifiers taking only the nearest neighbor ($k = 1$). This can be interpreted as a high accuracy considering the extremely unbalanced label distribution (see figure 2). Among the trained classifiers, we can also find some improvement when the trigram similarity weight ($\alpha_3$) is at least as high as the bigram similarity ($\alpha_2$). In addition, any KPCA embedding model beats all word2vec models for this task. For the sake of a fair comparison, the displayed word2vec results from Table 1 correspond to models where their vector size matches with a tested number of principal components $d$ of KPCA embeddings and tested with the same $k$ values for k-nearest neighbors. However, we also tested additional word2vec models with vector sizes up to 100 and up to 100 neighbors. None of these larger models reached a mean accuracy above 27%.

### 3.2 Splice junction recognition on DNA sequences

We encode DNA sequences from the dataset “Molecular Biology (Splice-junction Gene Sequences) Data Set” from UCI Machine Learning Repository [8]². This dataset consists of DNA subsequences represented as 30 characters out of the four nucleobases (A, T, C, G) plus other four characters (D, N, S, R) which mark ambiguity. The sequences may contain a splice junction between the 30 first and the 30 last characters. They are thus labeled with three different categories depending if they contain exon/intron boundary (EI class), intron/exon boundary (IE class) or neither (N). The distribution of the classes is displayed in Table 2.

We compute the similarity function considering all n-grams, giving all terms from (7) the same weight:

$$\alpha_i = \frac{1}{58}, \quad i \in \{2, \cdots, 59\}$$

$$\alpha_i = 0, \quad i \notin \{2, \cdots, 59\}$$

Using the learned representations, we train k-nearest neighbors classifiers to predict to which of the three classes each DNA sequence belongs to. Analyzing the prediction performance, as we can see from Table 3, we achieve a mean accuracy 94.67% with two different kernels. This result beats all baseline systems that are presented with the dataset, including knowledge-based artificial neural networks (KBANN) [11].

² https://archive.ics.uci.edu/ml/datasets/Molecular+Biology+(Splice-junction+Gene+Sequences)
<table>
<thead>
<tr>
<th>$d$</th>
<th>$\alpha_3$</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>$k = 5$</th>
<th>$k = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>76.76</td>
<td>65.19</td>
<td>65.20</td>
<td>63.02</td>
<td>60.56</td>
</tr>
<tr>
<td>5</td>
<td>0.75</td>
<td>76.67</td>
<td>66.10</td>
<td>65.94</td>
<td>63.74</td>
<td>60.85</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>76.56</td>
<td>65.73</td>
<td>65.78</td>
<td>63.33</td>
<td>60.50</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>76.72</td>
<td>65.33</td>
<td>65.14</td>
<td>63.01</td>
<td>60.27</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>77.01</td>
<td>65.19</td>
<td>65.14</td>
<td>62.64</td>
<td>59.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>76.80</td>
<td>66.01</td>
<td>66.38</td>
<td>64.54</td>
<td>61.92</td>
</tr>
<tr>
<td>10</td>
<td>0.75</td>
<td>76.72</td>
<td>66.13</td>
<td>66.83</td>
<td>64.30</td>
<td>61.95</td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
<td>77.17</td>
<td>66.45</td>
<td>66.70</td>
<td>64.36</td>
<td>61.96</td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td>77.23</td>
<td>66.30</td>
<td>66.78</td>
<td>64.40</td>
<td>62.43</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>77.13</td>
<td>66.26</td>
<td>66.67</td>
<td>64.42</td>
<td>62.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>77.28</td>
<td>66.45</td>
<td>66.96</td>
<td>64.82</td>
<td>62.60</td>
</tr>
<tr>
<td>15</td>
<td>0.75</td>
<td>77.05</td>
<td>66.12</td>
<td>66.66</td>
<td>64.44</td>
<td>61.90</td>
</tr>
<tr>
<td>15</td>
<td>0.5</td>
<td>77.23</td>
<td>66.94</td>
<td>66.17</td>
<td>64.01</td>
<td>62.00</td>
</tr>
<tr>
<td>15</td>
<td>0.25</td>
<td>76.82</td>
<td>65.15</td>
<td>65.51</td>
<td>63.48</td>
<td>61.11</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>76.51</td>
<td>65.52</td>
<td>65.34</td>
<td>63.76</td>
<td>60.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>77.51</td>
<td>66.16</td>
<td>67.00</td>
<td>65.06</td>
<td>62.67</td>
</tr>
<tr>
<td>20</td>
<td>0.75</td>
<td>76.93</td>
<td>65.95</td>
<td>66.00</td>
<td>64.23</td>
<td>62.22</td>
</tr>
<tr>
<td>20</td>
<td>0.5</td>
<td>77.05</td>
<td>65.01</td>
<td>65.29</td>
<td>63.76</td>
<td>61.61</td>
</tr>
<tr>
<td>20</td>
<td>0.25</td>
<td>76.67</td>
<td>64.85</td>
<td>64.94</td>
<td>63.10</td>
<td>61.22</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>76.43</td>
<td>64.88</td>
<td>64.70</td>
<td>62.76</td>
<td>60.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Mean accuracy in % predicting the verb tag with k nearest neighbors, trigram ratio $\alpha_3$ ($\alpha_2 = 1 - \alpha_3$) and d principal components applying different kernel functions. For the word2vec baselines, $d$ refers to the word vector size.
<table>
<thead>
<tr>
<th>Class</th>
<th>Nr. sequences</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>767</td>
<td>25%</td>
</tr>
<tr>
<td>IE</td>
<td>768</td>
<td>25%</td>
</tr>
<tr>
<td>Neither</td>
<td>1655</td>
<td>50%</td>
</tr>
</tbody>
</table>

Table 2: Class distribution of the splice-junction DNA sequences dataset

<table>
<thead>
<tr>
<th>d</th>
<th>k</th>
<th>1</th>
<th>3</th>
<th>6</th>
<th>10</th>
<th>14</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57.21</td>
<td>55.49</td>
<td>58.46</td>
<td>61.91</td>
<td><strong>62.54</strong></td>
<td>62.07</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>73.98</td>
<td>76.18</td>
<td>79.31</td>
<td>79.78</td>
<td><strong>81.03</strong></td>
<td>80.72</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>92.16</td>
<td>92.63</td>
<td>93.42</td>
<td>93.57</td>
<td><strong>94.51</strong></td>
<td>93.89</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>91.22</td>
<td>92.32</td>
<td>93.10</td>
<td>93.26</td>
<td><strong>93.42</strong></td>
<td><strong>93.42</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>90.44</td>
<td>92.63</td>
<td>93.10</td>
<td>93.26</td>
<td><strong>93.89</strong></td>
<td><strong>93.89</strong></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>89.66</td>
<td>92.79</td>
<td>93.42</td>
<td>92.95</td>
<td>93.57</td>
<td><strong>94.04</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>90.75</td>
<td>92.48</td>
<td>92.95</td>
<td>93.57</td>
<td><strong>94.51</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>90.13</td>
<td><strong>93.57</strong></td>
<td><strong>93.89</strong></td>
<td><strong>93.89</strong></td>
<td>92.95</td>
<td>93.42</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>90.13</td>
<td>91.22</td>
<td>92.95</td>
<td>93.73</td>
<td><strong>94.67</strong></td>
<td>93.57</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>89.50</td>
<td>91.69</td>
<td>92.16</td>
<td>92.95</td>
<td><strong>93.42</strong></td>
<td><strong>93.42</strong></td>
<td></td>
</tr>
</tbody>
</table>

(a) Polynomial kernel (degree 2)

<table>
<thead>
<tr>
<th>d</th>
<th>k</th>
<th>1</th>
<th>3</th>
<th>6</th>
<th>10</th>
<th>14</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52.66</td>
<td>56.11</td>
<td>59.09</td>
<td>58.78</td>
<td><strong>61.44</strong></td>
<td>60.19</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>70.06</td>
<td>75.24</td>
<td>74.92</td>
<td><strong>77.12</strong></td>
<td>76.18</td>
<td>76.49</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>90.60</td>
<td>90.44</td>
<td>90.28</td>
<td>91.38</td>
<td><strong>92.01</strong></td>
<td>92.32</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>92.79</td>
<td>93.42</td>
<td>94.04</td>
<td>93.26</td>
<td>93.26</td>
<td><strong>94.04</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>92.32</td>
<td>92.79</td>
<td>93.26</td>
<td>94.04</td>
<td>94.20</td>
<td>93.89</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>91.85</td>
<td>92.79</td>
<td>93.42</td>
<td><strong>93.89</strong></td>
<td>92.95</td>
<td>93.26</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>90.91</td>
<td>93.42</td>
<td>93.89</td>
<td><strong>94.04</strong></td>
<td>93.73</td>
<td>93.73</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>90.13</td>
<td>92.16</td>
<td><strong>93.42</strong></td>
<td>93.57</td>
<td>92.95</td>
<td>93.42</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>88.87</td>
<td>91.22</td>
<td>92.16</td>
<td>93.89</td>
<td><strong>94.67</strong></td>
<td>94.04</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>88.87</td>
<td>91.85</td>
<td>92.63</td>
<td><strong>93.26</strong></td>
<td>92.63</td>
<td>91.54</td>
<td></td>
</tr>
</tbody>
</table>

(b) RBF kernel ($\sigma = 0.72$)

Table 3: Mean accuracy in % predicting splice junctions with k nearest neighbors and d principal components applying different kernel functions. Several results beat the baseline system KBANN (93.68% mean accuracy).

4 Discussion and future work

We showed that our KPCA embedding approach to learn vector word representations can encode the morphology of the words in an unsupervised fashion, at least for the particular case of German verbs. The learned KPCA embeddings could beat by far any word2vec model in the task of predicting the grammatical tag. The highest accuracy was achieved by taking the nearest neighbor to predict the verb category. Due to this fact, we suspect that our proposed word representations tend to form clusters according to their word form, from which predicting a grammatical tag is a feasible task with simple classifiers such as k-nearest neighbors classifiers.

Since many NLP applications require also syntactic and semantic information about the words, good word embeddings should also incorporate information not only from the form of the represented word but also about the context in which they appear. In this direction, we will enhance our approach by adapting our similarity function so that it also considers the frequency of each evaluated word pair appearing in the same context or, alternatively, by taking our KPCA representations as input representation of a neural network architecture. For the latter, our KPCA would “just” substitute the one-hot encoding of most of neural
Fig. 3: Visualization of KPCA embeddings generated from the DNA sequences with RBF kernel, $\sigma = 0.72$ and seven principal components. Our dataset consists of DNA subsequences represented as 30 characters out of the four nucleobases (A, T, C, G) plus other four characters (D, N, S, R) which mark ambiguity. The sequences may contain a spline junction between the 30 first and the 30 last characters. They are thus labeled with three different categories depending if they contain exon/intron boundary (EI class), intron/exon boundary (IE class) or neither (N). Only their first three components are plotted. Best seen in color.

language models. Additionally, we will also extend our research evaluating the same approach on other morphologically rich inflected languages (like any of the Romance languages) or agglutinative languages (such as Turkish). We assume they may profit the most from our approach since their word forms reveal more grammatical information than word forms from more analytic languages such as English. To this extent, KPCA embeddings may also help to overcome the lack of linguistic resources of some of these non-English languages.

Furthermore, we presented how we can learn KPCA embeddings for DNA sequences. These representations proved to be useful in the task of predicting splice junctions. It would be also interesting to generalize our method to encode other types of discrete sequential data where also n-grams could be extracted, for instance text paragraphs (word n-grams), protein sequences (amino acid n-grams) or even sheet music (note n-grams).
References

jPL: A Java-based Software Framework for Preference Learning

Pritha Gupta, Alexander Hetzer, Tanja Tornede, Sebastian Gottschalk, Andreas Kornelsen, Sebastian Osterbrink, Karlson Pfannschmidt, and Eyke Hülsermeier

Intelligent Systems Group
Paderborn University

Preference learning (PL) is an emerging subfield of machine learning, which deals with the induction of preference models from observed preference information [3]. Such models are typically used for prediction purposes, for example to predict context-dependent preferences of individuals on various choice alternatives. Depending on the representation of preferences, individuals, alternatives, and contexts, a large variety of preference models and problems are conceivable.

We developed a software framework offering tools and algorithms for solving preference learning problems. While software frameworks for core machine learning problems such as classification abound, we are not aware of any comprehensive library of tools for preference learning. In fact, existing libraries are essentially restricted to one or two types of PL problems (e.g. [2], [6], [5], [4], [1]).

Our framework, called jPL, is implemented in Java. It is based on a unified data format, the Generic Preference Representation Format (GPRF), which is suitable for modeling data related to different kinds of preference learning problems. This also includes a dataset transformer, which converts data from several existing formats to GPRF. As problem classes, the framework currently supports collaborative filtering, instance ranking, label ranking, multilabel classification, object ranking, ordinal classification, and rank aggregation out of the box, with at least two algorithms being implemented for each problem. It provides a convenient command line interface as well as an API, both allowing one to configure the system using json files. The whole framework was developed in a quite generic way, so as to allow other problems and algorithms to be added easily.

Our framework also supports the evaluation and comparison of different methods in terms of standard validation techniques, and includes a set of commonly used loss functions. Just like the framework as a whole, the evaluation component is easily extensible by new evaluation techniques and loss functions.

References

1. V Dang. The lemur project-wiki-ranklib. lemur project.

1 https://github.com/intelligent-systems-group/jpl-framework
A Bayesian approach for comparing hypotheses about sequential data and its applications

Florian Lemmerich¹, Philipp Singer, Martin Becker², Lisette Espin-Noboa¹, Dimitar Dimitrov¹, Denis Helic³, Andreas Hotho², and Markus Strohmaier⁴

¹ GESIS - Leibniz Institute for the Social Sciences firstname.lastname@gesis.org
² University of Würzburg
³ Graz University of Technology
delic@tugraz.at
⁴ RWTH Aachen
markus.strohmaier@humtec.rwth-aachen.de

Abstract. Sequential data are found in many settings, e.g., as sequences of websites that users visit, as sequences of travel locations, or as sequences of songs a user listened to. To improve the understanding of the mechanisms that underlie the generation of such sequences, we developed a novel approach called HypTrails. It utilizes Bayesian hypothesis testing and first-order Markov chain models in order to enable the comparison of a set of hypotheses with respect to their plausibility considering some observed data. Each of the hypotheses captures a belief in transitions between the states. It can be derived from theory in the application domain, from other related datasets, or from human intuition. We applied this approach to study several phenomena in the online world, e.g., navigation behavior in Wikipedia or urban mobility data. In this talk, we want to give an introduction to HypTrails [4] and showcase selected real-world applications [3, 1, 2] that utilize it.

References

A Latent-Feature Plackett-Luce Model for Dyad Ranking Completion

Dirk Schäfer

dirk.schaefer@jivas.de

Abstract. Dyad ranking is a specific type of preference learning problem, namely the problem of learning a model that is capable of ranking a set of feature vector pairs, called dyads. In this paper a situation is considered where feature vectors consists of identifiers only. A new method based on learning latent-features is introduced for this purpose. The method is evaluated on synthetic data and is applied on the problem of ranking from implicit feedback.

Keywords: Preference Learning, Dyad Ranking, Plackett-Luce Model, Matrix Factorization, Implicit Feedback

1 Introduction

Preference learning is an emerging subfield of machine learning, which deals with the induction of preference models from observed or revealed preference information [3]. Such models are typically used for prediction purposes, for example to predict context-dependent preferences of individuals on various choice alternatives. Depending on the representation of preferences, individuals, alternatives, and contexts, a large variety of preference models are conceivable, and many such models have already been studied in the literature.

A specific problem within the realm of preference learning is the problem of label ranking, which consists of learning a model that maps instances to rankings (total orders) over a finite set of predefined alternatives [11]. An instance, which defines the context of the preference relation, is typically characterized in terms of a set of attributes or features; for example, an instance could be a person described by properties such as sex, age, income, etc. As opposed to this, the alternatives to be ranked, e.g., the political parties of a country, are only identified by their name (label), while not being characterized in terms of any properties or features.

In [9], dyad ranking is introduced as a practically motivated generalization of the label ranking problem. In dyad ranking, not only the instances but also the alternatives are represented in terms of attributes—a dyad is a pair consisting of an instance and an alternative. Moreover, for learning in the setting of dyad ranking, an extension of the Plackett-Luce model, a statistical model for rank data, was proposed. The approach was based on modeling utility scores of dyads via a bilinear product of feature vectors of instance and alternative respectively.
In this paper a problem is addressed which is in a sense diametrical to the above mentioned settings because instances and alternatives are not described by any attributes. The reasons of this circumstance could be that observable attributes do not relate well with the preference information or the attributes contain many missing values or they do not exist at all. Data preprocessing and feature engineering may fail in these cases. Often however, instances and alternatives are identified by a unique ID, especially when they are organized in a relational database. Otherwise training examples from a finite set can always be enumerated and a unique ID can be assigned. For this scenario a new variation of the bilinear Plackett-Luce model is introduced which is called Latent-Feature Plackett-Luce model (LFPL). In contrast to the original formulation of the bilinear Plackett-Luce model it can deal with rankings over dyads which are specified by identifiers only.

The rest of the paper is organized as follows. First a formal description of the dyad ranking completion problem is given in Section 2. The LFPL method to solve this problem based on latent-features is described in Section 3. The relation between dyad ranking completion and the application of learning to rank on implicit feedback data is described in Section 4. An overview of related methods is provided in Section 5. Experimental results are presented in Section 6, prior to concluding in Section 7.

2 Problem Setting

2.1 Dyad Ranking

The problem of a dyad ranking is to learn a model that accepts as input any set of (new) dyads and produces as output a ranking of them. A dyad is a pair of feature vectors \( z = (x, y) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \), where the feature vectors are from two (but not necessarily different) domains \( \mathcal{X} \) and \( \mathcal{Y} \).

Figure 1 shows the basic learning workflow: a learning algorithm creates a model from a finite set of training examples. The trained model can afterwards be used for producing rankings on (new) dyads. The training set \( \mathcal{D} = \{\rho_n\}_{n=1}^N \) consists of a set examples \( \rho_n (1 \leq n \leq N) \), where each example is a dyad ranking

\[
\rho_n : z^{(1)} \succ z^{(2)} \succ \ldots \succ z^{(M_n)}, \quad M_n \geq 2,
\]

of length \( M_n \), where \( M_n \) can vary from example to example.

2.2 Dyad Ranking Completion

The domain from which the training dyads and their members respectively stem from is denoted by \( \mathcal{R} \subset \mathcal{X} \) and \( \mathcal{C} \subset \mathcal{Y} \) (indicating (r)ows and (c)olumns). Depending on the occurrence of dyad members within the training set one can characterize the dyads which need to be ranked. Table 1 provides an overview of the types of new dyads which can be encountered at the prediction phase. Dyad ranking completion deals with dyads of type 1 which are dyads whose members
Fig. 1. Generic supervised learning task in dyad ranking.

have been encountered at the training phase individually but not yet jointly together. They define the gaps in the \( R \times C \) matrix and filling these by ranking all of them or subsets of them characterizes the task.

Table 1. Characterization of dyads depending on whether their members have been encountered in the training phase individually but not conjointly.

<table>
<thead>
<tr>
<th>Dyad Type</th>
<th>Member 1 Encountered?</th>
<th>Member 2 Encountered?</th>
<th>Prediction Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>( R \times C )</td>
</tr>
<tr>
<td>2</td>
<td>yes</td>
<td>no</td>
<td>( R \times Y )</td>
</tr>
<tr>
<td>3</td>
<td>no</td>
<td>yes</td>
<td>( X \times C )</td>
</tr>
<tr>
<td>4</td>
<td>no</td>
<td>no</td>
<td>((X \setminus R) \times (Y \setminus C))</td>
</tr>
</tbody>
</table>

Ranking with Identifiers Dyads whose members are not described by attributes can be utilized either by using existing database IDs or by assigning natural numbers to them. When these dyads are encountered at the training phase then any new dyads in the prediction phase must necessarily be of type 1.

3 A Plackett-Luce Model with Latent Features

3.1 Plackett-Luce Model

The Plackett-Luce (PL) model is a parameterized probability distribution on the set of all rankings over a set of alternatives \( y_1, \ldots, y_K \). It is specified by a parameter vector \( \mathbf{v} = (v_1, v_2, \ldots, v_K) \in \mathbb{R}^K_+ \), in which \( v_i \) accounts for the (latent)
utility or “skill” of the option $y_i$. The probability assigned by the PL model to a ranking $\pi$ is given by

$$P(\pi | v) = \prod_{i=1}^{K} \frac{v_{\pi(i)}}{v_{\pi(i)} + v_{\pi(i)+1} + \cdots + v_{\pi(K)}} = \prod_{i=1}^{K-1} \frac{v_{\pi(i)}}{\sum_{j=i}^{K} v_{\pi(j)}}.$$  (2)

Obviously, the Plackett-Luce model is only determined up to a positive multiplicative constant, i.e., $P(\pi | v) \equiv P(\pi | s \cdot v)$ for all $s > 0$.

As an appealing property of the PL model is that its marginal probabilities (probabilities of rankings on subsets of alternatives) are easy to compute and can be expressed in closed form. More specifically, the marginal of a PL model on $M < K$ alternatives $y_i(1), \ldots, y_i(M)$ is again a PL model with parameters $v_{i(1)}, \ldots, v_{i(M)}$.

**Bilinear Plackett-Luce Model** The bilinear model (BilinPL) \cite{9} is based on a functional formulation of the skill parameters as follows,

$$v(z) = v(x, y) = \exp(x^\top W y)$$  (3)

that is obtained by defining $\Phi$ as the Kronecker (tensor) product:

$$\Phi(x, y) = x \otimes y = (x_1 \cdot y_1, x_1 \cdot y_2, \ldots, x_r \cdot y_c) = \text{vec}(xy^\top),$$  (4)

which is a vector of length $p = r \cdot c$ consisting of all pairwise products of the components of $x$ and $y$, also known as cross-products. Thus, the inner product $\langle w, \Phi(x, y) \rangle$ can be rewritten as a bilinear form $x^\top Wy$ with an $r \times c$ matrix $W = (w_{i,j})$; the entry $w_{i,j}$ can be considered as the weight of the interaction term $x_i y_j$.

### 3.2 Latent-Feature based PL Model

To circumvent the limitation of the Bilinear PL model on the dependence of explicit features a novel extension of the Plackett-Luce model is proposed. It is based on latent-features and is called LFPL. For this model the PL log-skill parameter for a dyad $z = (i, j)$ is defined as $\log v(z) = U_i V_j^\top$.

The following notation is used. The indices $i$ and $j$ refers to the entities associated with the first and the second dyad members, whereas $U$ and $V$ are matrices with $U \in \mathbb{R}^{\lvert R \rvert \times K}$ and $V \in \mathbb{R}^{\lvert C \rvert \times K}$, where $K$ denotes the dimensionality of latent feature vectors. $\lvert R \rvert$ and $\lvert C \rvert$ denote the number of entities in $R$ and $C$ respectively. Both, $R$ and $C$ refer to special feature spaces which are one dimensional and are made up of natural numbers. The notation $U_k$ refers to the $k$–th row vector of the matrix $U$, whereas $U_{k,i}$ refers to the $i$–th latent factor of the $k$–th vector.

The conditional probability of observing a dyad ranking with the parameters $\theta = \{U, V\}$ can then be stated as:

$$P(\pi_n | \varrho_n, \theta) = \prod_{k=1}^{M_n-1} \frac{\exp \left[ U_{z_n(k,1)} V_{z_n(k,2)}^\top \right]}{\sum_{l=k}^{M_n} \exp \left[ U_{z_n(l,1)} V_{z_n(l,2)}^\top \right]}.$$  (5)
The function \( z_n(k, i) \) in the likelihood (5) refers to the \( i \)-th member of a dyad \( z \) observed in sample \( \varrho_n \), where \( i \in \{1, 2\} \). The index \( k \) refers to the dyad \( z_n^{(j)} \) which is put at the \( k \)-th rank of \( \pi_n \), s.t. \( \pi_n(k) = j \).

The model parameters \( \theta = \{U, V\} \) can be found by minimizing the negative log-likelihood (NLL) of the data,

\[
\ell = \sum_{n=1}^{N} \ell_n.
\]

In (6) the NLL of a single example (dyad ranking) is given by

\[
\ell_n = - \log P(\pi_n | \varrho_n, \theta)
\]

\[
= \sum_{k=1}^{M_n-1} \log \left( \sum_{l=k}^{M_n} \exp \left[ U z_n(l,1) V^T z_n(l,2) \right] \right) - \sum_{k=1}^{M_n-1} U z_n(k,1) V^T z_n(k,2). \tag{8}
\]

The overall objective which is to be optimized is hence of the form:

\[
\ell(\theta, D) + \Omega(\theta), \tag{9}
\]

where \( \Omega(\theta) \) is a regularization term for controlling the model capacity or likewise a countermeasure for preventing over-fitting. Note that the objective (9) is not convex and its minimization requires special considerations.

**Learning Algorithm** As learning algorithm the first order optimization technique alternating gradient descent is used. With this approach the model weights are updated by keeping one matrix \( U \) or \( V \) fixed respectively. The first derivatives are given by

\[
\frac{\partial \ell_n}{\partial U_{s,i}} = \sum_{k=1}^{M_n-1} \sum_{l=k}^{M_n} \frac{\mathbf{1}\{s = z_n(l,1)\} V_{z_n(l,2),i} \cdot \exp \left[ U z_n(l,1) V^T z_n(l,2) \right]}{\sum_{l=k}^{M_n} \exp \left[ U z_n(l,1) V^T z_n(l,2) \right]} - \sum_{k=1}^{M_n-1} \mathbf{1}\{t = z_n(k,2)\} U_{z_n(k,1),i}, \quad \tag{10}
\]

with \( s \in \mathcal{R}_n \) and \( 1 \leq i \leq K \). Here \( \mathcal{R}_n \) refers to the set of entities represented by numerical IDs that occur as first dyad members in the sample \( n \). For example, for a dyad ranking \( \rho_n: (x_3, y_2) \succ (x_3, y_{144}) \succ (x_5, y_9) \), \( \mathcal{R}_n \) would be \( \{3, 5\} \) and \( \mathcal{C}_n = \{2, 9, 144\} \). The derivatives for \( V \) can be stated similarly by treating the components of \( U \) as constants:

\[
\frac{\partial \ell_n}{\partial V_{t,i}} = \sum_{k=1}^{M_n-1} \sum_{l=k}^{M_n} \frac{\mathbf{1}\{t = z_n(l,2)\} U_{z_n(l,1),i} \cdot \exp \left[ U z_n(l,1) V^T z_n(l,2) \right]}{\sum_{l=k}^{M_n} \exp \left[ U z_n(l,1) V^T z_n(l,2) \right]} - \sum_{k=1}^{M_n-1} \mathbf{1}\{t = z_n(k,2)\} U_{z_n(k,1),i}, \quad \tag{11}
\]
with \( t \in C_n \) and \( 1 \leq i \leq K \).

Algorithm 1 uses these expressions together with the AdaGrad strategy for adaptive learning rate updates. AdaGrad provides the advantage of freeing the user from determining the initial learning rate values and update schedules while being simple to implement. The actual implementation utilizes a quadratic regularization term

\[
\Omega(\theta) = \frac{1}{2} \lambda (\|U\|_F^2 + \|V\|_F^2),
\]

(12)

where \( \| \cdot \|_F \) refers to the Frobenius matrix norm. This approach is theoretically justified by considering it as a problem of learning low rank matrices with a convex relaxation; a relaxation that is based on the nuclear (trace) norm regularization.

**Algorithm 1 LFPL Alternating Gradient Descent with AdaGrad**

Require: Initial matrices \( U \in \mathbb{R}^{|R| \times K}, V \in \mathbb{R}^{|C| \times K}, \) learning rate \( \gamma \), number of iterations \( T \), dyad rankings \( D_{Tr} = \{(\varrho, \pi)_n\} \).

1: \textbf{function} \text{train lfpl} (\( D_{Tr}, U, V, T \))
2: \hspace{1em} \textbf{for} \( i \leftarrow 1 \) to \( T \) \textbf{do}
3: \hspace{2em} \( \nabla_U \leftarrow \text{Formula (10)} \) on all samples \( (\varrho, \pi)_n \) \( 1 \leq n \leq N \)
4: \hspace{2em} \( H_U = H_U + (\nabla_U)^2 \)
5: \hspace{2em} \( U \leftarrow U - \gamma (H_U)^{-1/2} \odot \nabla_U \)
6: \hspace{2em} \( \nabla_V \leftarrow \text{Formula (11)} \) on all samples \( (\varrho, \pi)_n \) \( 1 \leq n \leq N \)
7: \hspace{2em} \( H_V = H_V + (\nabla_V)^2 \)
8: \hspace{2em} \( V \leftarrow V - \gamma (H_V)^{-1/2} \odot \nabla_V \)
9: \hspace{1em} \textbf{end for}
10: \textbf{return} \( (U, V) \)
11: \textbf{end function}

**3.3 Predictions with LFPL**

With LFPL, it is possible to solve the dyad ranking completion problem from Section 2.2 by making predictions involving dyads of type 1 (c.f. Table 1). They are all of the form \( z^{(k)} = (i_k, j_k) \), where \( i_k, j_k \in \mathbb{N} \) and both indices \( i_k \) and \( j_k \) appeared independently at other dyads during training but not jointly together yet. The most probable ranking can be predicted by calculating \( v^{(k)} = \exp(U_{i_k} V_{j_k}^\top) \) and then arranging the dyads in descending order to the scores.

\( ^1 \) Note, that any other learning rate update strategy for gradient descent could be used too.
3.4 Connection with the Bilinear PL Model

There are two ways to relate the LFPL model to the Bilinear PL model. The first relation can be seen by setting $W = UV^\top$. By describing each entity $i$ by means of a label embedding $e_i$ (e.g., via 1-of-$k$ encoding), it is possible to cast (5) to the Bilinear PL model formulation (3). The second relation can be seen by treating the latent feature vectors in the rows of $U$ and $V$ as object features and the bilinear weight matrix $W$ as identity matrix $I_K$. Both relations justify to classify the LFPL to be a bilinear model, too. But in contrast to BilinPL (Section 3.1) the LFPL model does not require the engineering of explicit features but learns latent-feature vectors under the same bi-linearity assumption instead. Without considering $W$ to be of lower rank it is not possible for the BilinPL approach to generalize to unseen dyad rankings of type 1 where dyad members are described by identifiers only.

4 Learning from Implicit Feedback

Implicit feedback is probably the most basic kind of preference statement. It refers to the selection of a single item from a set of choice alternatives at a time. A reasonable interpretation for that action is that the selected item is probably preferred over other items present at the time of selection. This kind of data can be represented in a rectangular schema of rows and columns, e.g., where they represent users and items. Cells in that schema indicate if an interaction between a particular row and a column happened. In what follows we denote row indices as $i_k \in R$ and column indices as $j_l \in C$. Formally, implicit feedback data is defined as the subset of indices $F \subset R \times C$. An index pair $(i, j)$ in $F$ can be interpreted as an action that happened between entities $i$ and $j$. The goal is to provide a ranking of dyads $(i, j)$ for which no interactions have been observed so far, i.e., $(i, j) \notin F$.

Without the loss of generality we turn now to the special case of the LFPL for contextual implicit feedback. Let the set of items on which interactions have been recorded for a context $i$ be denoted as $C_i^+ := \{ j \in C \mid (i, j) \in F \}$. We treat this scenario as a completion problem which we want to solve with dyad ranking. Figure 2 illustrates the conversion of original implicit feedback data to contextual dyad rankings. For each row $i_k$ (or user) a separate dyadic preference graph is generated. All graphs combined form a training data set $D_F$, which is

$$D_F = \{(i, j_a) \succ (i, j_b) \mid j_a \in C_i^+ \land j_b \in C_i \setminus C_i^+ \}.$$

To overcome the problem of over-fitting on high frequent items, the learning procedure needs a slight modification. The new procedure uses sampling from $D_F$ similar to BPRMF [7] but reuses the existing Algorithm 1 for this purpose. Instead of acting on single training examples it generates subsets of examples (batches) of size $L$. This new approach is called LFPL/IF (for implicit feedback) and is stated as Algorithm 2. As stopping criterion either the convergence of the performance on a hold-out set or a predefined maximal number of iterations can be used.
Algorithm 2 LFPL/IF Learning Algorithm

 Require: Implicit feedback data $D_F$, number of latent factors $K$, batch-size $L$
 1: Initialize $U \in \mathbb{R}^{|R| \times K}$, $V \in \mathbb{R}^{|C| \times K}$
 2: repeat
 3: $S \leftarrow$ draw $L$ many dyadic preferences from $D_F$
 4: Update LFPL $(S, U, V, 1) \triangleright$ Algorithm 1
 5: until Stopping Criterion
 6: return $U, V$

Gradients of LFPL/IF The calculations of the gradient matrices $\nabla U$ in Eq. (10) and $\nabla V$ in Eq. (11) of Algorithm 1 simplify drastically in the case of implicit feedback data. Let $j_a, j_b$ denote the indices of items, where the corresponding item $j_a$ is preferred over $j_b$. Let furthermore $i$ be the index of a user and $1 \leq k \leq K$ be a latent feature dimension. The NLL of a training example is then given by

$$\ell_n = \log(\exp(U_i V_{ja}) + \exp(U_i V_{jb})) - U_s V_{ja}.$$ 

And the derivatives can be calculated using following expressions:

$$\frac{\partial \ell_n}{\partial U_{i,k}} = V_{ja,k} \exp(U_i V_{ja}) + V_{jb,k} \exp(U_i V_{jb}) \exp(U_i V_{ja}) + \exp(U_i V_{jb}) - V_{ja,k} ,$$

(13)

$$\frac{\partial \ell_n}{\partial V_{ja,k}} = \frac{U_{i,k} \exp(U_i V_{ja}) \exp(U_i V_{ja}) + \exp(U_i V_{ja})}{\exp(U_i V_{ja}) + \exp(U_i V_{jb})} - U_{s,k} ,$$

(14)

$$\frac{\partial \ell_n}{\partial V_{jb,k}} = \frac{U_{i,k} \exp(U_i V_{ja}) + \exp(U_i V_{ja})}{\exp(U_i V_{ja}) + \exp(U_i V_{jb})} .$$

(15)

5 Related Work

Plackett-Luce Networks (PLNet) is a method for dyad ranking that is able to learn joint-feature representations [10]. Being based on a neural network it is
possible to express non-linear relationships between dyads and rankings. It is applicable of solving the dyad ranking completion problem too but it is not designed for the special case where dyad inputs are expressed by identifiers only.

Probabilistic Matrix Factorization (PMF) \[^8\] is a classical preference completion method for preferences in form of ratings. A substantial property of PMF is the learning and the prediction of quantitative instead of qualitative preferences by filling gaps of an incomplete rating matrix. Although LFPL and PMF differ fundamentally in this point, they also share some aspects. Firstly, both approaches are based on the same model class, i.e., matrices \( U \) and \( V \) are multiplied together to give a real valued matrix of dimensionality \( N \times M \). In both cases, higher matrix values indicate stronger preferences. Secondly, both approaches put Gaussian priors over \( U \) and \( V \) to control model capacity.

The methods BPRMF and WRMF are standard methods for learning on implicit feedback data. The LFPL method is closely related to BPRMF (the matrix factorization variant of Bayesian Personalized Ranking) although the first acts on dyad rankings of the form \((i, j_a) \succ (i, j_b)\) and the latter on triplets \(j_a \succ i \succ j_b\) \[^7\]. Both approaches share the same model class and also the assumption of Gaussian priors. LFPL generalizes BPRMF by supporting rankings of arbitrary lengths instead of pairwise preferences, i.e., rankings of length 2, and by providing contextual and non-contextual variants. WRMF is an approach that is also based on matrix factorization \[^6\]. It shares with LFPL and BPRMF the same model class but in contrast to them it is a pointwise approach (one item) instead of a pairwise approach (two items at a time).

6 Experiments

Ranking Completion (Synthetic Data) The advantage of LFPL over BilinPL and PLNet is shown for the problem of ranking completion with identifiers. Given a set of row entities \( R \) and a set of column entities \( C \), which are specified by their IDs only. Furthermore, provided with a training set of incomplete contextualized rankings over \( C \), where contexts are provided by entities from \( R \), the task is to create a model which can be used to predict arbitrary rankings for a set of new dyads in \( R \times C \).

One thousand rankings of lengths 10 were sampled from the LFPL model distribution. In a 10-fold CV experiment where 900 of them were used for training and 100 for testing for each run. The LFPL model used for the generation of rankings was determined by the matrices \( U \in \mathbb{R}^{|R| \times K} \) and \( V \in \mathbb{R}^{|C| \times K} \) which were generated by sampling from the normal distribution, with \( K = 5 \), the number of latent factors. To apply the dyads with identifiers on BilinPL and PLNet the natural numbers used as identifiers were converted into vector format using 1-of-k encoding. PLNet is configured to contain one hidden layer with 10 neurons. The results given in Table 2 confirmed the hypotheses that BilinPL and PLNet do not generalize well beyond the observed preferences. Especially PLNet is prone to overfitting. LFPL in contrast to BilinPL is able to share model
parameters under the assumption that preferences are representable with a lower rank matrix $W$.

**Table 2.** Results of the synthetic preference completion problem. Performance values in terms of Kendall’s $\tau$ are obtained via 10-fold CV on 1000 rankings, $|X_i| = 10$.

| $|R| \times |C|$  | Completeness (%) | BilinPL | LFPL | PLNet |
|-------------------|------------------|---------|------|-------|
| 40 x 50           | 2.026            | .844 ± .025 | .973 ± .006 | .721 ± .023 |
| 60 x 100          | 0.225            | .724 ± .018 | .950 ± .014 | .611 ± .031 |
| 80 x 150          | 0.056            | .770 ± .016 | .923 ± .009 | .493 ± .032 |

**Implicit Feedback (Real Data)** In this experiment the capabilities of LFPL/IF on implicit feedback data was tested. The data set Movielense (1M) was utilized due to its availability and high degree of popularity [5]. It is about 6040 users which provided a total amount of 1 million explicit ratings on 3900 movies. Similar to the experimental setup of [7] the rating data set was converted into implicit feedback data. Each rating $1 \leq r \leq 5$ is interpreted as a feedback and the task corresponds to estimating how likely a user would rate the residual entries.

For the evaluation we repeated the following leave-one-out procedure: One random rating entry per user is removed from the data. These entries are not considered for training but for benchmarking the trained model afterwards. Training and test sets are thus disjoint subsets of user associated item indices, i.e., $C_{i}^{Tr}, C_{i}^{Te} \subseteq C_{+}^{i}, 1 \leq i \leq |U|$ and $C_{i}^{Tr} \cap C_{i}^{Te} = \emptyset$. For evaluation of the models the average AUC is used:

$$AUC = \frac{1}{|U|} \sum_{u} \frac{1}{|E(u)|} \sum_{(j_a, j_b) \in E(u)} I[\hat{v}(u,j_a) > \hat{v}(u,j_b)] ,$$

in which the evaluation pairs $E(u)$ are given by

$$E(u) = \left\{ (j_a, j_b) \mid j_a \in C_{u}^{Tr} \land j_b \notin (C_{u}^{Tr} \cup C_{u}^{Te}) \right\} .$$

As baseline methods random guessing and the user-independent most popular approach are used. Most popular determines $\hat{v}(u,i)$ as the number of users that interacted with item $i$. The implementations of the BPRMF and WRMF methods were taken from the software package MyMediaLite 3.11 [4] and their default parameters were adopted. For LFPL the parameters were: dimension of latent feature space set to 10, regularization values were fixed to 0.01, number of iterations was 5000 and the batch size was set to 2000.

The results provided in Figure [3] reflect my expectation that LFPL/IF performs similarly as BPRMF and WRMF. All three methods outperform the base-
Fig. 3. Results of the Movielens 1M Implicit Feedback experiment.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC ± Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPRMF</td>
<td>0.926 ± 0.002</td>
</tr>
<tr>
<td>LFPL/IF</td>
<td>0.920 ± 0.001</td>
</tr>
<tr>
<td>Most Popular</td>
<td>0.860 ± 0.002</td>
</tr>
<tr>
<td>Random</td>
<td></td>
</tr>
<tr>
<td>WRMF</td>
<td>0.917 ± 0.002</td>
</tr>
</tbody>
</table>

lines methods to a large extent. In direct comparison LFPL/IF is slightly superior to WRMF and inferior to BPRMF in terms of AUC performance.

7 Conclusion

We presented a new dyad ranking approach based on the Plackett-Luce model with latent-feature vectors applicable for the dyad ranking completion problem. The advantage of LFPL is its applicability in scenarios where no information on dyad members are available but their unique identifiers. The problem of learning and prediction given implicit feedback data could be addressed as an application of dyad ranking completion. The learning algorithm is based on alternating gradient descent to deal with the non-convexity of the objective function. An obvious disadvantage is that predictions with latent features can be carried out in a straightforward way only on dyads within $\mathcal{R} \times \mathcal{C}$. LFPL in its current form is also a bilinear model and may be limited to model more complicated non-linear relationships between dyads and preferences.

References


Learning to Rank based on Analogical Reasoning

Mohsen Ahmadi Fahandar and Eyke Hüllermeier
Intelligent Systems Group
Paderborn University

Preference learning is a branch of machine learning dealing with the induction of preference models from observed preference information [2]. An important problem in the realm of preference learning is “learning to rank” in the setting of object ranking: On the basis of training data in the form of a set of rankings of objects (choice alternatives) represented as feature vectors, the goal is to learn a ranking function that predicts a linear order of any new set of objects [1].

In this paper, we propose a new approach to object ranking based on principles of analogical reasoning. More specifically, our basic line of reasoning is formalized in terms of so-called analogical proportions [3], and can be summarized by the following inference pattern:

\[
A \succ B, \quad A : B :: C : D \quad \Rightarrow \quad C \succ D
\]

Given four objects \(A, B, C, D\), if object \(A\) is known to be preferred to \(B\), and \(C\) relates to \(D\) as \(A\) relates to \(B\), then \(C\) (supposedly) preferred to \(D\).

Our learning method consists of two main building blocks: pairwise comparison and rank aggregation. Assuming training data in the form of pairwise preferences (longer rankings are broken into such pairs beforehand), and given a new set \(Q\) of query objects for which a ranking is sought, a weighted preference (which can be interpreted as a probability) is estimated for each pair \(C, D \in Q\). To this end, the number of preferences \(A \succ B\) and \(B \succ A\) are counted in the training data, where \(A\) and \(B\) are in analogical proportion to \(C\) and \(D\). In a second step, the (weighted) pairwise preferences are combined into an overall ranking. This is accomplished by means of suitable methods for rank aggregation.

Our first experimental results are promising. On data sets from various domains (sports, education, tourism, etc.), our approach turns out to be highly competitive to state-of-the-art methods for object ranking. Specifically strong performance is observed in situations where prediction requires a kind of knowledge transfer (for example, predicting a ranking of hotels in one city based on preferences for another city). The principle of analogical reasoning appears to be especially appropriate for this type of problems.

References

Semantic Query Processing: Estimating Relational Purity

Jan-Christoph Kalo¹, Christoph Lofi², René Pascal Maseli¹ and Wolf-Tilo Balke¹

¹ Institut für Informationssysteme, TU Braunschweig
r.maseli@tu-bs.de, {kalo, balke}@ifi.cs.tu-bs.de
² Web Information Systems Group, Delft University of Technology
c.lofi@tudelft.nl

Abstract. The use of semantic information found in structured knowledge bases has become an integral part of the processing pipeline of modern intelligent information systems. However, such semantic information is frequently insufficient to capture the rich semantics demanded by the applications, and thus corpus-based methods employing natural language processing techniques are often used conjointly to provide additional information. However, the semantic expressiveness and interaction of these data sources with respect to query processing result quality is often not clear. Therefore, in this paper, we introduce the notion of relational purity which represents how well the explicitly modelled relationships between two entities in a structured knowledge base capture the implicit (and usually more diverse) semantics found in corpus-based word embeddings. The purity score gives valuable insights into the completeness of a knowledge base, but also into the expected quality of complex semantic queries relying on reasoning over relationships, as for example analogy queries.

Keywords: Semantics of Relationships, LOD, Structured Knowledge Repositories, Word Embeddings

1 Introduction

To provide an intuitive and efficient user experience, future information systems need to offer powerful query capabilities with an awareness of the semantics of the query, as for example question answering systems [1] or intelligent digital assistants like MS Cortana, Google Now, or Apple Siri. Here, entities referred to in queries and relationships between those entities play a central role in the query processing process. Structured Knowledge Repositories like the Google Knowledge Vault [2] or WordNet [3] and Linked Open Data sources like DBpedia [4], Yago [5], usually serve as a premier source of such semantic information. However, due to their nature as structured knowledge bases, they are not always sufficient to implement some concepts required for intuitive queries: for example, information on human perception (like similarity or relatedness) or information on less clear attributes or fuzzy relationships are often omitted. As an example, consider the entities Brad Pitt and Angelina Jolie. In addition to their only (and outdated) DBpedia relationship spouse, their semantic relationship of
course is much more complex. There exists a plethora of additional relationships between them, which is quite hard to model: as for example, they co-acted in the same movies together, had many joint public appearances, and publicly split up again.

Here, word embeddings (such as recent skip-n-gram, neural embeddings [6, 7]) have been shown to provide an interesting additional source of semantic information on entities and their relationships: word embeddings learn a vector representation of words used in a large natural language corpus by exploiting the distributional hypothesis [8], thus promising to encode semantics based on the actual human perception and everyday use, implicitly provided by the language structure of the natural text corpus used for training (like news articles or encyclopedias). For example, it has been shown that the similarity of the resulting word vectors closely correlates with the perceived attributional similarity of their respective real-world entities [9], which allows for similarity queries but also for mapping between diverging user and knowledge base vocabulary for query processing in question answering [10] (i.e., when querying for “husband”, but the knowledge base only has information on “spouses”, similarity can help to suggest using spouse instead of husband).

An interesting use case of powerful semantic queries which were (re-)popularized by word embeddings are analogy queries. Using analogies in natural speech allows communicating dense information easily and naturally by implying that the “essence” of two concepts is similar or at least perceived similarly. Thus, analogies can be used to map factual and behavioral properties from one (usually better-known concept, the source) to another (usually less well known, the target) concept by exploiting both attributional and relational similarity. This is particularly effective for natural querying and explaining when only vague domain knowledge is available (e.g., “Okinawa is to Japan as is Hawaii to the US”). It has been argued [6] that such semantics can be expressed by simple vector arithmetics within the word embeddings space, however, it has also been shown that the performance of this type of analogy processing varies greatly with the type of relationships involved [11].

In this paper, based on contemporary computational analogy processing theory, we investigate the semantic relationship between vector arithmetics of word embeddings, the relationships found in structured knowledge repositories, and analogy semantics. The resulting contributions can be summarized as follows:

- We introduce the concept of purity scores for relationships in knowledge bases by investigating the vectors associated with each relationship in a word embeddings space. As word embeddings are based on rich language semantics of a large text corpus, we assume that they contain richer (or at least different) semantic information than structured knowledge bases, but only in implicit form. The purity score of a relationship represents the degree to which the knowledge base covers the implicit semantics of embeddings.
- We provide an extensive overview and examples of different relationships, their associated vectors, as well as their related source texts which have been involved in creating those vectors to clarify the concept of purity scores.
- We show that the popular analogy reasoning technique using word embedding vector arithmetics work well for pure relationships, while it does not work well for impure ones.
2 Related Concepts

In the following section, we revisit and summarize some of the core concepts underpinning our findings. This especially covers the general semantics of 4-term analogies, common word embeddings, and the offset method for analogical reasoning using vector arithmetics (as we already discussed in [12]).

Analogy and Relational Similarity. The semantics of analogies have been researched in depth in the fields of philosophy, linguistics, logic, and in cognitive sciences, such as [13–15]. However, those models are rather complex and hard to grasp computationally, and thus most recent works on computational analogy processing rely on the simple 4-term analogy model, which is given by two sets of word pairs (the so-called analogons), with one pair being the source and one pair being the target. A 4-term analogy holds true if there is a high degree of relational similarity between those two pairs. This is denoted by \([a_1, a_2] :: [b_1, b_2]\), where one relationship between \(a_1\) and \(a_2\) is similar to a relationship between \(b_1\) and \(b_2\), as for example in \([\text{US Dollar, USA}] :: [\text{Euro, Germany}]\). This model has several limitations, as is discussed in [16]: the semantics of “a high degree of relational similarity” from an ontological point of view is unclear as there can be plethora of relationships between the concepts of an analogon, but only some of them are of relevance for valid analogy semantics.

Therefore, we rely on an improved interpretation of the 4-term analogy model [16], and assume that there can be multiple relationships between the concepts of an analogon, some of them being relevant for the semantics of an analogy (the defining relationships), and some of them not. An analogy holds true if the sets of defining relationships of both analogons show a high degree of relational similarity. For illustrating the difference and importance of this change in semantics, consider the analogy statement: \([\text{Tokyo, Japan}] :: [\text{Braunschweig, Germany}]\). Tokyo is a city in Japan, and Braunschweig is a city in Germany, therefore both analogons contain the same “city is located in country” relationship, and this could be considered a valid analogy with respect to the simple 4-term analogy model. Still, this is a poor analogy statement from a human perspective because Braunschweig is not like Tokyo at all (therefore, this statement does neither describe the essence of Tokyo nor the essence of Braunschweig particularly well): the defining traits (relationships) of Tokyo in Japan should at least cover that Tokyo is the single largest city in Japan, and its capital. There are many other cities, which are also located in Japan, but only Tokyo has these two defining traits. Braunschweig, however, is just a smaller city in Germany, which might stand out for either its technical university or its historic city center (therefore, the defining relationships of both word pairs are not very similar). The closest match to a city like Tokyo in Germany should therefore be Berlin, which is also the largest city and the capital city.

Understanding which relationships define the essence of an analogon as perceived by humans is a very challenging problem, but this understanding is crucial for judging the usefulness and value of an analogy statement. Furthermore, the degree in which relationships are defining an analogon may vary with different contexts (e.g., the role of Berlin in Germany in a political discussion vs. the role of Berlin in Germany in a discussion about nightlife).
Word Embeddings, Relational Similarity, and Analogy Processing. Word embeddings represent each word in a predefined vocabulary with a real-valued vector, i.e. words are embedded in a vector space (usually with 50-600 dimensions). Most word embeddings will directly or indirectly rely on the distributional hypothesis [8] (i.e. words frequently appearing in similar linguistic contexts will also have similar real-world semantics), and are thus particularly well-suited to measure semantic similarity and relatedness between words (which is one of the foundation of the 4-term analogy definition), e.g., see [9]. In recent years, especially word embeddings relying on neural networks have become popular, with the skip-gram negative sampling approach (SGNS) [6, 7] being one of the best known examples.

The straight-forward application of word embeddings is computing similarity between two given words [9] by measuring the cosine similarity. However, many (but not all) word embeddings show some very interesting and surprising additional property: it seems that not only the cosine distance between vectors represents a measure for similarity and relatedness of the embedded words, but that also the difference vectors between word pairs implicitly represent the relationships between two entities, and thus carry analogy semantics [17]. For example, the difference between the vector for “man” and “king” seems to represent the concept/relationship of being a ruler, and the closest word vector to “woman” plus the “ruler” concept vector will be “queen” (see Fig. 1; this method is also sometimes called the offset method).

To a certain extent, these semantics can be attributed to the distributional hypothesis: in natural speech, concepts carrying similar semantics will frequently co-occur in similar context. Therefore, the difference vector should implicitly encode the defining relationships between two concepts as discussed in the previous section (i.e.: Tokyo/Japan and Berlin/Germany will likely occur in similar contexts in natural speech, while Braunschweig/Germany will likely appear in different context and will thus have a different difference vector). The ability of word embeddings to perform this analogical reasoning process has been evaluated using several standardized test sets (see next section), but is still not well understood and can fail quite often, which is related to our introduced purity score.

In a more formal fashion, a word embedding can be used to solve analogy completion queries as follows [6]: Given the query \([a_1, a_2] :: [b_2, ?]\), the word embedding provides the respective word vectors \(\vec{a}_1\), \(\vec{a}_2\), and \(\vec{b}_1\). Then, the vector \(\vec{b}_2\) representing the query’s solution can be determined by finding the word vector in the trained vector

![Fig. 1. Schematic representation from our GloVe Word Embedding Vectors for the DBpedia relationship country.](image)
space $V$ which is closest to $\vec{a}_2 - \vec{a}_1 + \vec{b}_1$ with respect to the cosine vector distance, i.e. 

$$
\vec{b}_2 = \arg \max_{\vec{x} \in V, \vec{x} \neq \vec{a}_2, \vec{x} \neq \vec{b}_1} (\vec{a}_2 - \vec{a}_1 + \vec{b}_1)^T \vec{x}.
$$

**Relational Benchmarking.** The Mikolov Benchmark set [18] is one of the most popular benchmark sets for testing the analogy reasoning capabilities of word embeddings, covering 19,558 4-term analogies. However, only 14 distinct relationships are covered, and most of them (9) focus on grammatical properties the relationship “is plural for a noun”, e.g., [mouse,mice]:[dollar,dollars] or “is superlative”. Five relationships are of a semantic nature (i.e. “is capital city for country” [Athens,Greece]:[Oslo,Norway], “is currency of country”, “city in state”, “male-female version” (including the often cited [king,queen]:[man,woman]). The test set is generated by collecting pairs of entities which are members of the selected relationship either manually or from Wikipedia and DBpedia, and then combining these pairs into 4-term analogy tuples. For example, for the “city in state” relationship, 68 word pairs like [Dallas,Texas] or [Miami,Florida] are collected, and then combined by a cross product. The related Wordrep dataset [19] extends the Mikolov set by adding more challenges, and expanding to 25 different relationships.

For the Mikolov dataset, the authors showed that skip n-gram word embeddings [6] using the offset method could solve analogy completion queries (i.e. [Athens,Greece]:[Oslo,?]) with an accuracy of 53.3% overall, 50% for semantic relationships (like ‘capital of’), and 55.9% for syntactic ones (like ‘plural of’). No deeper analysis of the relationships for which this technique performs well was provided. However, this was analyzed in more detailed in [11] using subjective feedback on relational similarity from human users. Here, the authors identified as the core problem which hinders the offset methods for analogy reasoning the presence of multiple relationships between two entities which influence human perception, as some relationships are perceived more dominant than others, e.g., quite often, the relationship intended in the Mikolov analogy challenge (e.g., “city in state”) was not perceived as dominant as some other relationship perceived by human subjects (e.g., “home of best football team in state”). While this argument is formulated slightly differently, those experimental results strongly support the hypothesis of defining relationships [16] discussed in the previous sections. Also, different perception of relationships introduces problems with symmetry or transitivity of relational similarity not holding from a user’s perspective [11]. A similar result is also supported by experiments in [20].

Based on the intuition obtained in those experimental results, in the following, we define the concept of *relational purity* approximating in how far a relationship given in an analogy challenge is indeed perceived as the relevant or defining relationship, which gives us insights both into the its suitability for analogy query processing, but can also serve as an indirect and implicit measure for the semantic completeness of a knowledge base with respect to that relationship.
3 Purity Score for Relationships

In the following section, we further explain our idea of relationship purity and provide a formal definition of the concept. We generalize the idea to relationships between entities, and support our findings with an analysis of DBpedia relationships.

Motivation. As we motivated, the semantic relationships modelled in triple format (consisting of subject, predicate, object triples) in state-of-the-art knowledge bases are often a stark simplification of the relationships between the respective entities as perceived by humans. As a result, many relationships are left out (e.g., Angelina Jolie and Brad Pitt having a public fight about their children), or several related but still perceptually different relationships are generalized and grouped into a single relationship. In Figure 2, we have visualized a 2-dimensional scaling of a GloVe embedding trained on Wikipedia for the “spouse” relationship (i.e., the difference vectors between two spouses). The more parallel those relationship vectors are, the more similar their representation is in the embedding space, and therefore, the more similar their captured semantics should be (as the texts in which those relationships are discussed share similar contexts). In the case of the spouse relationship, we observe that the relationship vectors of different couples are quite diverse: However, we can observe that the vectors of the two actor couples (Jolie/Pitt and Heard/Depp), are more similar, since they are linked by more than one single relationship.

If, in DBpedia, we look at the entities Tolkien and The Hobbit, we observe that they are connected by only a single relationship instance: Tolkien is the author of The Hobbit. In the text found in Wikipedia, a whole paragraph is used to describe the relationship between the respective entities: how he initially wrote The Hobbit for his children, how he never planned to publish it, how his friends liked it and pressured him towards publication, etc. Similarly, the same “is author of” relationship is used to link Max Frisch to his novel Homo Faber, whereas the Wikipedia text offers a much deeper insight into their relationship than the structured knowledge base does (see Fig. 2 for a visualization).

However, as defined by the DBpedia ontology, the recommended use of the “is author of” relationship is much more general and it can connect any type of author with their work of any kind (this can be novels, scientific papers, screenplays, music, paintings and even software programs). Thus, it generalizes the semantics of a larger amount of diverse subrelationships to one single relationship, not capturing their perceived semantic differences anymore (i.e., Pablo Picasso authoring the painting “Les Demoiselles d'Avignon” is perceived quite different than Max Frisch authoring “Homo Faber” by most). However, we believe that this loss of semantics can be modelled and captured by analyzing rich textual corpora. Here, for the “is author of” relationship, we claim that the rich diversity of different types of author relationship leads to a low purity score, while other relationships like “is currency of country” have high purity scores (while there might be diverse relationships between countries and their currencies, this diversity is comparably low). This purity, i.e., diversity of usage of a relationship can be observed in word embeddings.

As a further example relationship in Fig. 2, consider the “is in country” relationship for cities: the relationship between Braunschweig and Germany for example (i.e., there
is nothing particularly unique about Braunschweig), is quite different from the relationship from Paris to France. Paris is not only a city located within France, but also the country’s largest city, the location of the French government and its capital. The relationship vector of Rome to Italy and London to UK look very similar.

Please note that all those vectors have 300-dimensions, and that thus such simple visual analytics are quite crude due to the loss of semantics when mapped to the 2D-plane using Principal Component Analysis. Hence, we introduce a formal definition of relational purity in the next section.

**Computing the Purity Score.** Word embeddings were shown to provide linear sub-structures that can represent implicit relational similarity of all relationships between two entities as having similar (i.e., having a high cosine similarity) difference vectors between their word vectors. This characteristic is mainly used for analogical query processing using the offset method. However, we can adopt a similar notion to compute the *purity score* of relationships. Given a set subject entities $S$ and object entities $O$, which are connected by a relationships $R$ in a structured knowledge base, we compute

![Fig. 2. Three DBpedia relationships in the embedding space visualized in 2D using Principal Component Analysis.](image-url)
the purity score of the relationship $R \subseteq S \times O$ as the standard deviation from the average difference vector of the entities. Given a triple $(s, r, o)$, its relationship vector in a word embedding is defined by the respective entities difference vector: $\vec{r} = \vec{s} - \vec{o}$. We define the average vector for relationship $R$, given the set of relation vectors $\vec{R}$ as $\vec{a}_R = \frac{\sum_{\vec{r} \in R} \vec{r}}{|R|}$. Now, we define the purity score of a relationship as the standard deviation of the cosine distance from every relationship instance vector to the average vector $\vec{a}_R$.

The cosine distance between two vectors is defined as $\cos(\vec{u}, \vec{v}) = 1 - \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||_2 ||\vec{v}||_2}$. Note that negative similarity leads to distances between 1 and 2 in case the vectors are directed in opposite directions. The purity of a relationship $R$ is now defined as:

$$pu_r(R) = 1 - \frac{\sum_{\vec{r} \in R} \cos(\vec{a}_R, \vec{r})^2}{|R|}$$

A high variance in the directions of the relationship vectors (represented by cosine distances to the average vector), results in impure relational information in the embedding, i.e. low purity scores. Similarly, a low variance in the directions (so parallel relationship vectors) lead to a high purity score.

4 Evaluation

In this section, we introduce our experimental setup, and specifically focus on how we represented DBpedia entities and relationships in a corpus-based word embedding. Afterwards, we compute the purity scores for DBpedia relationships and visualize and discuss some examples to obtain a better intuition of the results.

**Experimental Setup.** We extracted relationships between entities from the largest Linked Open Data (LOD) data store DBpedia [4], a knowledge graph that is built by extracting knowledge from Wikipedia Infoboxes. As result, our dataset covers 18 million unique relationship instances between entities from around 1,200 relationships as defined by the DBpedia ontology, ranging from capital city relationships, over causal relationships, to biological relationships between living organisms.

As a training corpus for our word embedding, we downloaded a dump of the English version of Wikipedia from 01/2017. For linking DBpedia entities and relationships to the embedding, as a first step, we performed named entity recognition and disambiguation on the text corpus using DBpedia Spotlight [21] and replace the recognized entities by their respective DBpedia URI. For creating the word embedding, we use GloVe [7] and train multiple models with varied window size as described in the original GloVe paper between 8 and 15. Similar to their results [7], we found out that semantic relationships are better represented in the embedding when we use the larger window size. Furthermore, we ensured that the sliding window does not reach over different articles and sentence boundaries, preventing words appearing in wrong contexts. We varied the number of embedding dimensions between 50, 100 and 300, finding that DBpedia relationships are represented best by choosing 300 dimensions. Thus, the fol-
Following results all are based on a GloVe embedding with window size 15, 300 dimensions and 100 training iterations. The minimum word frequency is set to 8. Our corpus and embeddings are available on request.

**Fig. 3.** Two-dimensional visualization of randomly sampled difference vectors for 9 DBpedia ontology relationships. The projection was computed using principal component analysis. The relationships are ordered by purity from top-left to bottom-right. Further information on the relationships can be seen in Table 1.

**Result Visualization.** For visualizing relationship vectors and their purity, we selected 9 DBpedia relationships from very pure to extremely impure relationships from a 300-dimensional GloVe embedding. We used principal component analysis to scale the results to two dimensions. The results are visualized in Figure 3. Subject entities are visualized by a red dot, object entities by a blue triangle and the difference vector, representing the relationship between the entities, is visualized by a green line. The relational similarity between the relationships given by cosine distance of the difference vectors. Hence, parallel vectors have cosine similarity 0, whereas orthogonal vectors have similarity 1. (Due to the two-dimensional scaling of the original vectors, the cosine
similarity is not perfectly represented in the visualization.) Pure relationships (top left) have highly parallel relationship vectors, whereas impure relationships (bottom right) are very diverse. Since impure relationships have nearly no parallel relationship vectors, the offset method does not lead to meaningful results. Hence, most analogy queries based on this method return incorrect results.

Table 1. DBpedia relationships, their purity score and the domain and range of the respective relationships.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Purity</th>
<th>Range, Domain</th>
<th>Relationship</th>
<th>Purity</th>
<th>Domain, Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>kingdom</td>
<td>0.94</td>
<td>Species, Taxonomic Rank</td>
<td>currency</td>
<td>0.56</td>
<td>Country, Currency</td>
</tr>
<tr>
<td>mediaType</td>
<td>0.93</td>
<td>Book, Media Type</td>
<td>musicComposer</td>
<td>0.53</td>
<td>Composition, Composer</td>
</tr>
<tr>
<td>domain</td>
<td>0.92</td>
<td>Species, Taxonomic Rank</td>
<td>musicalBand</td>
<td>0.49</td>
<td>Song, Music Band</td>
</tr>
<tr>
<td>gender</td>
<td>0.90</td>
<td>Person, Gender</td>
<td>director</td>
<td>0.47</td>
<td>Movie, Director</td>
</tr>
<tr>
<td>phylum</td>
<td>0.89</td>
<td>Species, Taxonomic Rank</td>
<td>award</td>
<td>0.46</td>
<td>Person, Award</td>
</tr>
<tr>
<td>timeZone</td>
<td>0.82</td>
<td>City, Timezone</td>
<td>coach</td>
<td>0.46</td>
<td>Sports Team, Coach</td>
</tr>
<tr>
<td>country</td>
<td>0.81</td>
<td>City, Country</td>
<td>capital</td>
<td>0.43</td>
<td>Country/State, Capital</td>
</tr>
<tr>
<td>sport</td>
<td>0.80</td>
<td>Organization, Sport</td>
<td>writer</td>
<td>0.41</td>
<td>Book/Screenplay, Writer</td>
</tr>
<tr>
<td>nationality</td>
<td>0.80</td>
<td>Person, Nationality</td>
<td>writer</td>
<td>0.38</td>
<td>Book/Movie, Editor</td>
</tr>
<tr>
<td>profession</td>
<td>0.79</td>
<td>Person, Job</td>
<td>editor</td>
<td>0.36</td>
<td>Book/Painting/Soft-ware/Movie, Author</td>
</tr>
<tr>
<td>deathCause</td>
<td>0.79</td>
<td>Person, Cause of Death</td>
<td>author</td>
<td>0.24</td>
<td>Person, Person</td>
</tr>
<tr>
<td>campus</td>
<td>0.77</td>
<td>University, Campus</td>
<td>album</td>
<td>0.23</td>
<td>Person, Person</td>
</tr>
<tr>
<td>occupation</td>
<td>0.76</td>
<td>Person, Job</td>
<td>child</td>
<td>0.09</td>
<td>Person, Person</td>
</tr>
<tr>
<td>religion</td>
<td>0.75</td>
<td>Person, Religion</td>
<td>doctoralAdvisor</td>
<td>0.07</td>
<td>Person, Person</td>
</tr>
<tr>
<td>instrument</td>
<td>0.74</td>
<td>Person, Instrument</td>
<td>parent</td>
<td>0.04</td>
<td>Person, Person</td>
</tr>
<tr>
<td>hometown</td>
<td>0.67</td>
<td>Music Artist, City</td>
<td>partner</td>
<td>0.01</td>
<td>Person, Person</td>
</tr>
<tr>
<td>mayor</td>
<td>0.67</td>
<td>City, Person/Political Party</td>
<td>partner</td>
<td>0.01</td>
<td>Person, Person</td>
</tr>
<tr>
<td>party</td>
<td>0.66</td>
<td>Person, Political Party</td>
<td>relative</td>
<td>0.01</td>
<td>Person, Person</td>
</tr>
<tr>
<td>university</td>
<td>0.58</td>
<td>Person, University</td>
<td>spouse</td>
<td>0.01</td>
<td>Person, Person</td>
</tr>
</tbody>
</table>

Purity of DBpedia Relationships. We have evaluated the purity of all DBpedia ontology relationships for which we could find at least two relationship instances in our embedding. This resulted in more than 400 different relationship embeddings with different purity scores. In Table 1, we show an excerpt covering the complete purity spectrum. Particularly pure are relationships with only a few different subjects or objects, as for example, the biological kingdom relationship from DBpedia which connects species to one of six different biological kingdoms. The country relationship linking cities to their countries or states also has a high purity score of 0.81; most of the relationships instances are parallel, having some exceptions as shown in Figure 3. The author relationship, as already discussed in Section 3, has a purity of 0.36. Since its domain comprises entities of very different type, the resulting relationship vectors show only few similarities. However, we can see several different clusters of similar (parallel) relationships, indicating that the relationship is impure (i.e., each cluster represents a subrelation which is not modelled directly in the knowledge base).

The spouse relationship connecting two married persons is the most diverse (and thus impure) relationship in our dataset, having a purity score of only 0.01. This diversity has several reasons: On the one hand, this relationship is symmetric (which is not covered by the default similarity measurement using cosine distance), therefore the vectors for man and his wife is directed in the opposite direction to the vector that con-
nects a woman to her husband. Furthermore, the persons being married and their relationships to each other are quite different from couple to couple (see our introductory examples) which is well represented in text but not in a knowledge base.

5 Summary and Future Work

In this paper, we investigated the semantic interplay between explicit relationships modelled in structured knowledge bases like DBpedia, and their representation in corpus-based word embeddings. While word embeddings do not explicitly represent individual relationship instances between two entities, the implicit representation of all relationship instances as a difference vectors between two word/entity vectors can potentially cover a much wider range of (perceptual) semantics based on the corpus used for training. This notion is usually exploited for embedding-based relational similarity and analogy queries. However, it can also be used to shed some light on the nature of a specific relationship and how well it is represented in a knowledge base. To this end, we introduced the concept of relational purity, which implicitly represents how uniform the usage of a give relationship is in natural text. This results on several interesting observations: some relationships (like “is spouse of”) carry much richer semantics in their textual representations (e.g., while DBpedia just contains that Brad Pitt and Angelina Jolie are/used to be married, texts talking about both indicate a large number of additional potential and currently not covered relationships) indicated by a low purity score, while hierarchical relationships from the Linné taxonomy for animal and plant life are very pure – e.g., indicating that there is exactly one specific semantic relationship between two entities in the taxonomy (for example, squirrels are rodents – there are no significant other relationships mentioned in natural text connecting “squirrel” with “rodent”).

For the future, we plan to exploit our insights obtained in this work for improving complex semantic query processing (by e.g., being able to give an assessment of the potential reliably of an answer based on reasoning over relationships), but also for designing processes for uncovering potential semantic gaps in knowledge bases and mining for missing information in a targeted fashion.

References

Leveraging Patient Similarity Analytics in Personalized Medical Decision Support System

Araek Tashkandi$^1$ and Lena Wiese$^2$

$^1$ King Abdullah University, Faculty of Computing and Information Technology, 21589 Jeddah, Kingdom of Saudi Arabia. asatashkandi@kau.edu.sa
$^2$ Georg-August-University Goettingen, Institute of Computer Science, Goldschmidtstr. 7, 37077 Goettingen, Germany. wiese@cs.uni-goettingen.de

1 Abstract

Patient similarity analytics harness the information wealth of electronic medical record (EMR) for supporting medical decision making. Finding a group of patients having similar features (for example, similar lab results or similar diagnoses), helps medical staff with treatment decisions or health predictions. There are different approaches for patient similarity metrics (PSMs) as described by [3]. We implemented patient similarity for mortality prediction as [2]. The method is based on cosine similarity that exploits similarities between ICU patients along multiple dimensions. We applied the proposed method on a real-world EMR data set MIMIC-III [1] containing both demographic data (like gender) as well as lab results. SQL is mainly used for implementing the similarity calculation. Our computation and analysis are conducted in MonetDB. Based on the requirement of calculating our PSM by SQL, our hypotheses is that column-oriented database management systems will outperform the row-oriented ones. To test this assumption, we conducted the same analyses in PostgreSQL.

We intend to use other PSMs that are mentioned by [3] and compare them to ours. For enhancing the computational efficiency, various technologies will be considered. Since “pairwise PSM computation is very much parallelizable” [2], such technologies are the big data analytic platforms such as Apache Hadoop. We will conduct an evaluation of the different implementations of patient similarity algorithms along with the selected technologies to achieve an efficient prediction and computation combination.

References

De-Anonymisierungsverfahren: Kategorisierung und Anwendung für Datenbankfragen
De-anonymization: Categorization and use-cases for database queries

Johannes Goltz, Hannes Grunert, and Andreas Heuer
Universität Rostock, Lehrstuhl für Datenbank- und Informationssysteme, Institut für Informatik, 18051 Rostock

Abstract: The project PArADISE deals with activity and intention recognition in smart environments. This can be used in apartments, for example, to recognize falls of elderly people. While doing this, the privacy concerns of the user should be kept. To reach this goal, the processing of the data is done as close as possible at those sensors collecting the data. Only in cases where the processing is not possible on local nodes the data will be transferred to the cloud. But before transferring, it is checked against the privacy concerns using some measures for the anonymity of the data. If the data is not valid against these checks, some additional anonymizations will be done.

This anonymization of data must be done quite carefully. Mistakes might cause the problem that data can be reassigned to persons and anonymized data might be reproduced. This paper gives an overview about recent methods for anonymizing data while showing their weaknesses. How these weaknesses can be used to invert the anonymization (called de-anonymization) is shown as well. Our attacks representing the de-anonymization should help to find weaknesses in methods used to anonymize data and how these can be eliminated.


Keywords: Datenbanken, Datenschutz, (De-)Anonymisierung
1 Einleitung


Erschwerend kommt hinzu, dass Nutzer häufig die Datenschutz- oder Nutzungsvereinbarung aus Gründen der Bequemlichkeit nicht mehr lesen und so keine Ahnung haben, wie ihre Daten weiterverarbeitet werden. Beispielsweise versprachen einige Nutzer in einem Experiment durch die Nutzung eines öffentlichen Hotspots ihr erstgeborenes Kind oder liebstes Haustier dem Hoster des Hotspots [12]. Dies zeigt, dass grundsätzlich eine große Diskrepanz zwischen der Aufklärungspflicht des Anbieters und der Bereitschaft der Nutzer, diese zu lesen, besteht.

Bei Nutzung von aktuellen Anonymisierungsverfahren muss allerdings die Implementierung genau betrachtet werden, da kleine Fehler fatale Auswirkungen auf das Resultat haben können. Zu eng gewählte Randbedingungen für eine Anonymisierung von einem Datenbestand können beispielsweise dazu führen, dass die ursprünglichen Daten rekonstruiert werden können, oder zumindest teilweise wieder personenbeziehbare Daten offenlegen werden.


2 Das PArADISE-Projekt

Die Forschung an der Universität Rostock beschäftigt sich unter anderem interdisziplinär mit Assistenzsystemen. Hierbei sollen zum Beispiel Stürze in Wohnungen erkannt werden. Es wird neben der Sensorik auch die Datenverarbeitung untersucht. An dieser Stelle kommt PArADISE zum Einsatz,

1 Privacy Aware Assistive Distributed Information System Environment (PArADISE)
welches die Prinzipien von \textit{Privacy by Design} umsetzt, indem die Implementierung von datenschutzfördernden Techniken (Privacy Enhancing Technologies, PETs) erfolgt. Dabei werden im Speziellen die rechtlichen Anforderungen nach Datensparsamkeit und Datenvermeidung durch Techniken zur \textit{Anfrageumschreibung} umgesetzt. Ausgehend von dem reduzierten Datenbestand werden verschiedene \textit{Anonymisierungstechniken} verwendet, um das Ergebnis der Anfrage datenschutzkonform zu veröffentlichen. In diesem Artikel wird beschrieben, wie durch De-Anonymisierungstechniken überprüft wird, ob der scheinbar anonymisierte Datensatz wieder deanonymisiert werden kann. Ziel der Überprüfung ist die Reduzierung von Angriffsmöglichkeiten innerhalb der Verarbeitungskette im PArADISE-Framework.

\textbf{Privacy by Design durch Anfrageumschreibung}


\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{paradise-framework.png}
\caption{Schichtaufbau des PArADISE-Frameworks}
\end{figure}
Privacy by Design durch Daten-Anonymisierung

Sollten Daten an höhere Schichten weitergegeben werden, so werden diese zusätzlich mit den hinterlegten Richtlinien verglichen. Sobald zu viele Informationen enthalten sind, wird eine Anonymisierung durchgeführt.

Dazu müssen wir einerseits die zu kontrollierenden Anonymitätsmaße und ihre Parameter festlegen, andererseits aber auch Verfahren implementieren, die dieses Anonymitätsmaße auf dem in die Cloud zu übertragenden Datenbestand effizient berechnen können (siehe folgendes Kapitel [3]). Um zu testen, wie sicher die Anonymität des Nutzers gewährleistet ist, entwickeln wir gleichzeitig Angriffsverfahren (De-Anonymisierung), die Schwachstellen in der Anonymisierung aufdecken sollen (siehe Kapitel [4]).

3 Anonymisierungsverfahren und -maße

Um die Anonymisierung von Datenbeständen automatisieren zu können, werden entsprechende Maße benötigt, die den aktuellen Grad der Anonymität bestimmen. Sollten die vorliegenden Daten noch nicht anonym genug sein, können Algorithmen genutzt werden, um den Informationsgehalt zu verringern. Dies wird so lange iterativ in Schritten durchgeführt, bis ein entsprechendes Maß erfüllt ist. Dieser Absatz beschreibt entsprechende Methoden zum Messen des Grades der Anonymisierung. Ein Kern-Bestandteil ist dabei der Quasi-Identifikator [1].

Definition 1 Ein Quasi-Identifikator (QI) $Q_T$ ist eine endliche Menge von Attributen $\{A_1, \ldots, A_j\}$ einer Tabelle $T$ mit einer endlichen Menge von Attributen $\{A_1, A_2, \ldots, A_n\}$. Hierbei gilt $\{A_1, \ldots, A_j\} \subseteq \{A_1, A_2, \ldots, A_n\}$. Mit Hilfe des QIs ist es möglich, mindestens ein Tupel der Tabelle $T$ eindeutig zu bestimmen [2]. Eine Menge von Tupeln $t$ von $T$, welche bezüglich des QIs $Q_T$ nicht unterscheidbar sind, wird als $q^*$-Block bezeichnet.


Ein sogenanntes „sensitives Attribut“ ist ein Attribut, das nicht mit personenbeziehbaren Informationen in Verbindung gebracht werden darf, da dies der entsprechenden Person schaden könnte. Beispielsweise könnte dieses Attribut die Diagnose in einer Tabelle sein, in der Patientendaten mit entsprechenden Diagnosen abgespeichert sind (siehe Tabelle 1). Während die Informationen der
Spalte *Diagnose* allein nicht problematisch sind, werden sie in Verbindung mit Name und Vorname durchaus kritisch. Die Mengen der sensitiven Attribute und der Attribute von QIs und Schlüsseln sind nicht zwangsläufig disjunkt. Es kann daher vorkommen, dass jedes Attribut Teil eines QIs ist.

### 3.1 Anonymisierungsmaße

Die im Folgenden vorgestellten Maße für die Anonymität einer Relation lassen sich vor allem in Kombination mit der Technik der Generalisierung und Unterdrückung einsetzen. Diese werden im weiteren Verlauf vorgestellt.

#### k-Anonymität

Die *k-Anonymität* stellt die geringsten Anforderungen an die zu bewertenden Daten. Der Wert *k* gibt dabei an, wie viele Tupel es mit jeweils gleichem QI geben muss. Eine formale Definition ist in [9] zu finden.

![Fig. 2. Einordnung der Deanonymisierung in das PArADISE-Framework](image)

Je nach Wert für *k* und dem QI müssen die Daten, sollten sie aktuell nicht die geforderte k-Anonymität erfüllen, verallgemeinert werden. Dafür kann sehr gut die Generalisierung genutzt werden. Der Vorgang wird dabei iterativ so lange wiederholt, bis eine ausreichende Anonymisierung durchgeführt wurde. Beispielhaft ist dies in Tabelle 1 gezeigt.

#### l-Diversität und t-Closeness

*l-Diversität* und *t-Closeness* stellen Verschärfungen der k-Anonymität dar. *l-Diversität* nimmt sich der Problematik an, dass der Attributwert des sensitiven Attributes eines *q*-Blocks für jedes Tupel darin gleich sein könnte. Angenommen der Attributwert *Röteln* sei in Tabelle 1 ebenfalls *Diabetes*, dann würde die Tabelle damit immer noch k-Anonymität für *k=2* erfüllen, allerdings keine
l-Diversität für $l=2$ mehr. Der Wert $l$ gibt entsprechend an, wie viele unterschiedliche Werte für das sensitive Attribut im entsprechenden $q^*$-Block auftauchen müssen [9].

Bei $t$-Closeness wird die Verteilung der Attributwerte des sensitiven Attributes in Bezug zur Verteilung der Attributwerte in der gesamten Relation betrachtet. Die Verteilung darf dabei pro $q^*$-Block höchstens um $t$ von der Gesamtverteilung abweichen [9]. Eine Herausforderung dieses Verfahrens ist die Messung der Verteilung der Werte. Während dies bei numerischen Attributwerten einleuchtend und vergleichsweise einfach erscheint, wird es bei abstrakten Werten komplizierter. Hier bieten sich die Kullback-Leibler- oder auch die Jensen-Shannon-Divergenz an [14]. Für $t$ gilt, im Gegensatz zu $k$ und $l$, je kleiner desto anonymer werden die Daten. Typischerweise liegt der Wert für $t$ zwischen 0 und 1.

**Differential Privacy**


**3.2 Anonymisierungsverfahren**


**Generalisierung**

Die Generalisierung ist hierbei ein spaltenorientiertes Verfahren. Es werden zu generalisierende Attribute ausgewählt, anschließend alle Attributwerte dieser Spalten (die Domäne) auf ein entsprechendes Intervall abgebildet. Die originalen Werte einer Tabelle bilden die Grunddomäne, welche auf weitere Domänen generalisiert wird [13].
Unterdrückung


<table>
<thead>
<tr>
<th>Zeile</th>
<th>Alter</th>
<th>Diagnose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>Diabetes</td>
</tr>
<tr>
<td>2</td>
<td>84</td>
<td>Fraktur des Beins</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>Blutkrebs</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>Inkontinenz</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>Röteln</td>
</tr>
</tbody>
</table>

Zeile Alter Diagnose

<table>
<thead>
<tr>
<th>Zeile</th>
<th>Alter</th>
<th>Diagnose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10-19</td>
<td>Diabetes</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>3</td>
<td>20-29</td>
<td>Blutkrebs</td>
</tr>
<tr>
<td>4</td>
<td>20-29</td>
<td>Inkontinenz</td>
</tr>
<tr>
<td>5</td>
<td>10-19</td>
<td>Röteln</td>
</tr>
</tbody>
</table>

Table 1. Beispieltabelle (original links, generalisiert und unterdrückt rechts) - Zur Vereinfachung wurden nur zwei Spalten genutzt. Der QI sei das Alter, die Diagnose das sensitive Attribut. Durch Unterdrückung konnte der Wert der Spalte Alter auf ein Intervall von 10 abgebildet werden und die Tabelle erfüllt k-Anonymität für \( k=2 \) und l-Diversität für \( l=2 \) (q*-Blöcke wurden farblich hervorgehoben).

In der rechten Relation von Tabelle 1 ist zu erkennen, wie Generalisierung und Unterdrückung arbeiten. Zeile 2 wurde unterdrückt, da das Alter einen stark abweichenden Wert im Verhältnis zu den anderen Werten darstellt. Um trotz des Wertes k-Anonymität für \( k=2 \) zu erfüllen, hätten die Werte sonst auf ein entsprechend großes Intervall abgebildet werden müssen und alle Werte hätten mit großer Wahrscheinlichkeit im selben Intervall gelegen. Die einzelnen q*-Blöcke wurden zusätzlich farblich hervorgehoben. Sie unterscheiden sich bezüglich des QIs nicht. Ist bekannt, welches Alter die entsprechende Person hat, so ist nicht mehr ersichtlich, welche Diagnose ihr gestellt wurde.

Slicing

Ein weiteres Verfahren wird als Slicing bezeichnet. Hierbei wird eine Relation \( R \) in \( m \) vertikale und \( n \) horizontale Teilrelationen aufgeteilt. Innerhalb dieser Teilrelationen werden die Tupel zufällig sortiert, bevor alle Teilrelationen wieder zu einer kompletten Relation zusammengefügt werden [10]. Es ist zu beachten, dass unbedingt angegeben werden muss, an welchen Stellen in der Relation die Trennung vorgenommen wurde. Zusammenhängende Auswertungen zwischen Attributen, die in unterschiedlichen Teilrelationen standen, sind nicht mehr

4 De-Anonymisierungsverfahren

Wir beschreiben nun, an welchen Stellen die vorher vorgestellten Anonymisierungsverfahren versagen. Es lassen sich grundsätzlich zwei unterschiedliche Ansätze unterscheiden. Zum einen kann lediglich die Anfrage zur De-Anonymisierung von Daten betrachtet werden, zum anderen ist auch eine De-Anonymisierung auf Grundlage der vorliegenden Daten möglich.

4.1 Anfragebasierte De-Anonymisierungsverfahren


4.2 Datenbasierte De-Anonymisierungsverfahren

Bei datenbasierten Verfahren wird lediglich auf die aus der Auswertung erhaltende Ergebnisrelation einer Anfrage geachtet, und nicht auf die Anfrage an sich. Hier kommen die im vorangegangenen Abschnitt vorgestellten Anonymisierungsmaße zum Einsatz, um den Grad der Anonymität zu bestimmen. Diese weisen allerdings Schwachstellen auf, die Beachtung finden müssen.


Je nach veröffentlichten Daten kann auch die Sortierung der Tupel dem Angreifer helfen, persönliche Daten aus Ergebnissen zu extrahieren. Grundsätzlich sind Ergebnisrelationen immer sortiert. Dies liegt an den internen Speicherstrukturen der Datenbanksysteme [15]. Sollten allerdings mehrere Veröffentlichungen der gleichen Daten mit unterschiedlichen Quasi-Identifikatoren gemacht werden, so kann es zum Problem kommen, dass diese Daten eventuell einfach über die Sortierung verknüpft werden können. In Tabelle 2 ist dies beispielhaft zu sehen. Ähnlich verhält es sich, wenn der Angreifer einen direkten Zugang zur Datenbank nutzen kann. Damit könnte er die gleiche Anfrage mehrfach stellen und so hoffen,
dass vom System unterschiedliche Attribute der Quasi-Identifikatoren gewählt werden und so die Anonymisierung unterschiedlich umgesetzt wird. Zusätzlich könnte es auch passieren, dass eventuell ein anderer Quasi-Identifikator gewählt wird. Das Problem lässt sich allerdings auch sehr leicht beheben, indem die Ergebnisrelation einfach vor der Veröffentlichung zufällig sortiert wird.

<table>
<thead>
<tr>
<th>Geburtsjahr</th>
<th>Postleitzahl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>18055</td>
</tr>
<tr>
<td>1983</td>
<td>18057</td>
</tr>
<tr>
<td>1965</td>
<td>18055</td>
</tr>
<tr>
<td>1963</td>
<td>18055</td>
</tr>
<tr>
<td>1975</td>
<td>18059</td>
</tr>
<tr>
<td>1977</td>
<td>18057</td>
</tr>
<tr>
<td>1955</td>
<td>18181</td>
</tr>
</tbody>
</table>

Table 2. Ursprungstabelle (links) und jeweils eine der Spalten anonymisiert, sodass k-Anonymität für k=2 erfüllt ist. Quasi-Identifikator ist Geburtsjahr und Postleitzahl. Die Originaltabelle lässt sich durch direktes nebeneinanderlegen rekonstruieren.


5 Automatisierung eines Angriffs


Dies würde sehr viel Arbeit ersparen, ist aktuell aber nur mit äußerst großem Aufwand realisierbar. Eine Hilfestellung für die Wahl des richtigen Angriffvektors hingegen kann durch vergleichsweise einfache Techniken erreicht werden.
Durch eine statistische Auswertung der Ergebnisse kann ein schneller Überblick über die vorliegenden Daten gewonnen werden.


Sollten Werte, welche für die Bestimmung der statistischen Daten benötigt werden, aus der Datenbank abgefragt werden, könnte es zu Problemen kommen, wenn sich in der Zwischenzeit der Datenbestand verändert hat, oder auch die Ausgabe für jede Anfrage eventuell anders anonymisiert wurde. Weiterhin wurde in PArADISE eine Möglichkeit geschaffen, Wertebereiche der einzelnen Spalten einschränken zu können, um so fehlerhafte beziehungsweise nicht relevante Werte aus der statistischen Berechnung ausschließen zu können (siehe [5]).

Durch eine automatisierte Generalisierung können die Attributwerte des sensitiven Attributes so lange iterativ generalisiert werden, bis für jeden q*-Block ein eindeutiger Wert zugeordnet ist. Dabei müssen Duplikate nach jeder Iteration gelöscht werden. Mit den Informationen ist es einem Angreifer anschließend möglich, einen allgemeineren, aber immer noch möglichst spezifischen, Wert zu erkennen, ohne dass er aktiv einschreiten muss.

6 Zusammenfassung

Grundsätzlich lässt sich sagen, dass trotz Anonymisierung die Daten nie zu 100 Prozent sicher vor einem Angriff sind. Allerdings kann die Möglichkeit der De-Anonymisierung durch Angreifer sehr stark verringert werden. Auf der anderen Seite muss geprüft werden, ob die anonymisierten Daten noch für die nötigen Auswertungen ausreichend Informationen enthalten. Es sollte ein Maximum für die Werte der Anonymisierungsmaße gewählt werden, sodass gerade noch genügend Informationen für die gutartigen Anfragen enthalten sind, die von einem Assistenzsystem für die Analyse erlaubter Aktivitäten (wie Stürze) benötigt werden. Die Ausführung von bösertigen Anfragen, etwa die Ableitung genauerer Nutzerprofile oder Bewegungsprofile, die nicht zur Analyse der erlaubten Aktivitätserkennung beitragen, sollten dagegen verhindert werden.

Das Schichtkonzept des PArADISE-Frameworks [7] bietet eine sehr gute Voraussetzung für die Anonymisierung von Daten. Es kann einfach differenziert werden, wohin die Daten weiter gereicht werden und wie stark sie entsprechend anonymisiert werden müssen. Die trotz des Schichtkonzeptes in die Cloud zu übertragenden Daten, die für den Anbieter des Assistenzsystems erforderlich sind, um die Aufgaben des Assistenzsystems erfüllen zu können, müssen dann schlussendlich mit den in diesem Artikel vorgestellten Verfahren (a) auf Anonymität geprüft, (b) eventuell weiter generalisiert und gefiltert, und (c) durch die automatische Generierung von Angriffen auf Schwachstellen geprüft werden. Durch die Kombination von anfrage- und datenbasierten Verfahren für die De-Anonymisierung hoffen wir aber, in PArADISE ein höchstmögliches Niveau an Privatheit des Nutzers bewahren zu können (siehe auch [7]).

Danksagung

Wir danken den anonymen Gutachtern für ihre konstruktiven Kommentare.

Literaturverzeichnis

8. Grunert, H., Heuer, A.: Rewriting complex queries from cloud to fog under capability constraints to protect the users’ privacy. Open Journal of Internet Of Things 3(1), 31–45 (2017), proceedings of the International Workshop on Very Large Internet of Things in conjunction with the VLDB 2017 Conference in Munich, Germany.


Einsatzmöglichkeiten von Crowdsourcing im Umfeld des Mecklenburgischen Flurnamenarchivs
Applications of Crowdsourcing in The Mecklenburg Field Name Archive

Martin Lichtwark, Holger J. Meyer and Andreas Heuer

University of Rostock, 18051 Rostock, Germany,
Martin.Lichtwark@uni-rostock.de, Holger.Meyer@uni-rostock.de, Andreas.Heuer@uni-rostock.de

Abstract. The benefits of crowdsourcing is depicted in the context of the Mecklenburg Field Names Archive. This archive encompasses a large collection of field names. Transforming the hand-written field name lists and the corresponding map entries into structured digital content is still a manual task. This paper presents insights into crowd based realizations that have been evaluated in the project WossiDiA. Afterwards, we present the ongoing crowd based process of transferring field name lists and corresponding map entries into a hypergraph database. We conclude comparing our approach to related work.


Schlüsselwörter: Digitales Archiv, Crowdsourcing, WossiDiA, Flurnamenarchiv, Transkription, Verortung
1 Motivation


2 Das Mecklenburgische Flurnamenarchiv


Die noch vorhandenen echt-realen Restbestände des FNA wurden im Zuge eines, von der Deutschen Forschungsgemeinschaft (DFG) geförderten, Digitalisierungsprojekts.
sierungsprojektes in den Bestand von WossDiA\textsuperscript{1} integriert [11]. Bis ins Jahr 2010 lagen diese lediglich in Form einer Access-Datenbank vor, werden jedoch im Rahmen des hier vorgestellten Projektes in eine Graph-Datenbank, die eine Vernetzung im räumlich-zeitlichen Kontext erlaubt, überführt [12, 10].

3 Herausforderungen und das Potenzial von Crowdsourcing

Auch wenn das Anwendungspotenzial von Crowdsourcing (CS) vielseitig ist und in den unterschiedlichsten Bereichen bereits zum Einsatz kommt, ist gerade im Zusammenhang zum Bestand des Mecklenburgischen Flurnamenarchivs der interdisziplinäre Teilbereich \textit{Digital Humanities} im Kontext dieser Initiative als besondere Spezialisierung, unter anderem durch die Zusammenarbeit mit einem externen CS-Dienstleister zur vollständigen Abarbeitung, hervorzuheben.

Die Aufbereitung und Analyse des Archivs im Rahmen der Konzeption für die eigentliche Umsetzung ergab zwei zu lösende Kernprobleme, auf die es sich zu fokussieren galt.

Abb. 1: Fragebogen der Flurnamenforschung — Beispielort: Boissow; Quelle: Wossidlo-Archiv

Entscheidend und zielführend für eine erfolgreiche Umsetzung sind hierbei die Teilbereiche der Überarbeitung (Transkription) der Flurnamen selbst und der Verortung der eingezeichneten Nummern der Fragebögen der Flurnamenforschung (Abb. 1) des analogen Archivs, in die sich die Kernprobleme der Deskription des FNA klassifizieren lassen [1]. Eben diese Teilaufgaben stellen die Crowd-basierte Abarbeitung vor diverse Herausforderungen. Diese sind

\textsuperscript{1} \url{https://www.wossidia.de/}
nicht ausschließlich rein technischer Art, hinzu kommen Aspekte wie die Akqui-
rierung und Motivation der Crowdworker, die Qualität der Abarbeitung selbst 
sowie weitere Qualitätsfaktoren, wozu unter anderem Vollständigkeit, relative 
und absolute Genauigkeit, aber auch Korrektheit und Konsistenz zählen.

**Teilaufgaben** Auf Basis des spezifischen Bestandes des FNA und der Art der 
Erfassung ist eine Zweiteilung der Aufgabenschwerpunkte unabdingbar. So ist 
eine Transkription der Namen als erster Schritt notwendig. Hierbei gibt es vor 
allen Herausforderungen in der Vielfältigkeit der Typografie und der Diversität 
von Typen von Handschriften, welche Expertise benötigen, um einen qualita-
tiv genügenden Prozess zur Digitalisierung dieser Namen möglich zu machen. 
Gleichermaßen ist in diesem Zusammenhang auch eine Mitführung der Numme-
rierung zur Referenzierung der Namen mit den Markierungen in den Karten für 
den nachfolgend notwendigen Schritt der Verortung entscheidend. Diese Num-
merierungen werden erst nach dem Ende der Georeferenzierung ihre Bedeutung 
verlieren.

Für die Verortung stellt die Diversität an Karten, wie Messtischblätter, Flur-
karten, Übersichtskarten, Gutskarten oder angefertigte Handzeichnungen, um 
nur einige zu nennen und die Vielfalt an Formaten in Form von unterschiedli-
chen Maßstäben eine Herausforderung für die Crowd-basierte Abarbeitung dar. 
Technologisch kann hier auf bekannte Web Mapping Services wie die von Go-
gle Maps und OpenStreetMap zur Realisierung zurückgegriffen werden. Zudem 
bietet sich aufgrund der vielfältigen Unterschiede eine Visualisierung mittels ei-
er Side-by-Side Darstellung an. Die zu bearbeitende Diversität im Bestand des 
FNA verdeutlicht die nachfolgende Zusammenstellung in Abb. 2.

![Abb. 2: Diversität von Karten und handgeschriebenen Flurnamen; Quelle: Wossidlo-
Archiv](image-url)
Ziel der vorangegangenen wissenschaftlichen Forschungsarbeit war es zum einen die Herausforderungen und Probleme einer möglichen Abarbeitung aufzuzeigen und andererseits gleichermaßen Lösungsmöglichkeiten durch bereits etablierte Plattformen oder auch Anbieter von solchen Umsetzungsprozessen zu evaluieren.


<table>
<thead>
<tr>
<th>Nr.</th>
<th>Anbieter</th>
<th>Link</th>
<th>Kategorie</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amazon Mechanical Turk</td>
<td><a href="http://www.mturk.com">www.mturk.com</a></td>
<td>Mikrotasks</td>
</tr>
<tr>
<td>2</td>
<td>Clickworker</td>
<td><a href="http://www.clickworker.de">www.clickworker.de</a></td>
<td>Mikrotasks</td>
</tr>
<tr>
<td>3</td>
<td>CrisisCommons</td>
<td><a href="http://www.crisiscommons.org">www.crisiscommons.org</a></td>
<td>Engagement</td>
</tr>
<tr>
<td>4</td>
<td>CrowdFlower</td>
<td><a href="http://www.crowdflower.com">www.crowdflower.com</a></td>
<td>Mikrotasks (Meta)</td>
</tr>
<tr>
<td>5</td>
<td>Crowdguru</td>
<td><a href="http://www.crowdguru.de">www.crowdguru.de</a></td>
<td>Mikrotasks (Meta)</td>
</tr>
<tr>
<td>6</td>
<td>Crowdsource</td>
<td><a href="http://www.crowdsource.com">www.crowdsource.com</a></td>
<td>Mikrotasks (Meta)</td>
</tr>
<tr>
<td>7</td>
<td>Fiverr</td>
<td><a href="http://www.fiverr.com">www.fiverr.com</a></td>
<td>Mikro- bis Makrotasks</td>
</tr>
<tr>
<td>8</td>
<td>Fixmystreet</td>
<td><a href="http://www.fixmystreet.com">www.fixmystreet.com</a></td>
<td>Engagement</td>
</tr>
<tr>
<td>9</td>
<td>Georeferencer</td>
<td><a href="http://www.georeferencer.com">www.georeferencer.com</a></td>
<td>Mikrotasks</td>
</tr>
<tr>
<td>10</td>
<td>Samasource</td>
<td><a href="http://www.samasource.org">www.samasource.org</a></td>
<td>Mikrotasks</td>
</tr>
</tbody>
</table>

Tab. 1: Übersicht untersuchter Plattformen und Anbieter

Weiterhin können, im Falle einer mittel- bis langfristigen Orientierung für ein ergänzendes internes Projekt zum Ausbau des Bestandes in Form einer Eigenentwicklung, nachfolgende Anbieter einen interessanten Beitrag bezüglich angewandter Implementierungen, Verfahrensweisen und den Umgang mit Crowdworkern bieten: CrisisCommons, Fixmystreet und ebenfalls Georeferencer. Somit beherbergten fünf der untersuchten Anbieter einen potenziellen Mehrwert zum Erreichen der Zielstellung des gestellten Problems im Rahmen des FNA. Die anderen in der Untersuchung selektierten Anbieter sind aufgrund der Machbarkeit,
des eingeschränkten oder unvollständigen Portfolios, auf Basis wirtschaftlicher Aspekte oder durch ausschließende rechtliche Bedingungen nicht einsetzbar.

4 Arbeiten zur Umsetzung und Integration in WossiDiA

4.1 Aktuelle Umsetzung


Eine Übersicht zum Prozess zeigt die nachfolgende Abb. 3. Deutlich wird hierbei die vollständige Auslagerung der Teilaufgaben inklusive aller abhängigen Rahmenbedingungen, wie beispielsweise eine Qualitätssicherung durch Validatoren oder abschließende Sichtkontrollen je Teilaufgabe, an das beauftragte Unternehmen. Neben regelmäßigen Gesprächsterminen mit Crowd Guru zur Abstimmung des Fortschritts, konnte das Unternehmen anfangs aus einem Kollektiv von 100 Crowdworkern schöpfen, welche sich mit der ersten Teilaufgabe der Transkription beschäftigen sollten. Diese wurden zunächst durch eine initiale Testaufgabe, in Form einer Übersetzung eines einfachen in Sütterlin geschriebenen Rezeptes einer Eierpfanne mit Champignons, identifiziert. Aufgrund des speziellen Bestandes und der Diversitäten an Handschriften und Typografien kristallisierten sich allerdings 40 fähige Personen heraus, welche den komplexen Anforderungen in vollem Umfang gerecht werden konnten, denn wo bei einem Rezept noch Zusammenhänge erschlossen werden können, ist dies bei einer Transkription von separat stehenden Flurnamen nicht möglich. Bestätigt und bewertet wurde die Ergebnismenge im Rahmen durchgeführter Qualitätsprüfungen, welche wiederum durch zwei identifizierte Experten innerhalb der Crowdworker vorgenommen
wurde. So wurden innerhalb von anderthalb Wochen etwa 3100 Einheiten (Sei-
ten) des gelieferten Bestandes abgearbeitet. Bei den verbleibenden Einheiten gab es jedoch zusätzliche Probleme aufgrund der Lesbarkeit und der Typografie, sodass am Ende noch einmal 100 Einheiten neu abgearbeitet werden mussten, durch 10 bis 15 spezialisierte Crowdworker, die auf Grundlage ihrer vorherigen Transkriptionsergebnisse speziell dafür selektiert werden konnten. Mit solchen Herausforderungen war, nach der Aufarbeitung und Untersuchung des Archivs und der Kernproblematiken im Rahmen der Konzeption, jedoch zu rechnen.


In annähernd zwei Wochen wurden somit insgesamt 3906 Einheiten als digitalisierte Seitenobjekte im ersten Schritt, der Transkription der Namen mit Nummern, verarbeitet.

Anschließend startete mit den vorher geprüften Transkriptionen die Veror-
4.2 Integration in WossiDiA


Für einige folgt eine kurze Beschreibung und die zugehörigen Datendefinitionsanweisungen. Der Knotentyp \textit{Flur} repräsentiert Informationen zum Flurnamen selbst, Namen, die intern verwendete Nummer in der liste und Karte, die ermittelte Ortslage und eventuelle erfasste Bemerkungen des Beiträgers:

\begin{verbatim}
CREATE NODETYPE Flur (  
  Nummer String,  (: Number in the list; related to the place :)  
  Name String,  (: Name of the recorded cadastral section :)  
)
\end{verbatim}
Abb. 4: Flurname, Mappe und Beiträger als Hyperkanten modelliert

Lokation String, (: Location; cardinal direction :)  
Info String (: Remarks for the field name :)

Die 1:m Zuordnung von Flurnamen zu Orten erfolgt über einen speziellen Hyperkantentyp **Flurnamenverortung**:

```
CREATE EDGETYPE Flurnamenverortung MODEL (
  Ort:place IN,
  Flurname:Flur+ OUT
);
```

Während die speziell die Definition von Hyperkanten viele Ähnlichkeiten mit den semi-strukturierten Inhaltsmodellen von XML aufweist, lehnt sich die Graphanfragesprache GrafL stark an XQuery, neben der allgemeinen Form der Klauseln insbesondere bei der Form der Variablenbindung und der Verwendung von Knoten- und Kantenkonstruktoren.

5 Verwandte Arbeiten


6 Zusammenfassung und weiterführende Arbeiten

Für die Übertragung der Flurnamen und deren Position, aus den zumeist in Kurztext- oder Sütterlinschrift vorliegenden Listen und Karten in Form von Messblättern, lokalen Flurkarten oder Handzeichnungen, wurde ein Crowdsource-

Die voranschreitenden Vorgänge und Forschungen im Umfeld von WossiDiA halten zudem viel Potenzial für eine tiefenenschließende Verzahnung der verschiedenen Bestände bereit. So sind nicht nur Vernetzungen der Flurnamen kulturell wertvoll, sondern auch denkbare Querverweise zu anderen Beständen wie den Ortschroniken, um nur einen momentan stark bearbeiteten Bereich des Instituts für Volkskunde (IVK) zu nennen.

Zudem möglich wäre für die Erfassung von Georeferenzierungen eine flächendeckende Darstellung Mecklenburgs in Form einer Landeskarte, welche durch ein Ampelschema erschlossene, teilerschlossene und noch zu erschließende Gebiete offerieren könnte. Denkbar wäre auch an dieser Stelle eine Verzahnung mit diversen anderen Beständen, wie Orte, Gebiete der Erzählung etc., welche zukünftig ebenfalls Geodaten beherbergen könnten.

Eine solche Darstellung und Vernetzung von Beständen könnte potenziell auch mit dem sogenannten Holsten-Archiv für Pommern [7], welches sich aktuell in Stettin befindet, mittels weiterführenden Gemeinschaftsprojekten und Forschungsarbeiten vorgenommen werden, denn der „Lehrer und Volkskundler [., Robert Holsten,] tat in Pommern das, was Richard Wossidlo in Mecklenburg machte: Er sammelte regionale Bezeichnungen, Sitten und Gebräuche, die ihm seine ≪Zuträger≫ aus allen Kreisen Pommerns überbrachten.“ [5].


Angedacht vom IVK ist eine solche kooperative Zusammenarbeit, sodass es bereits mehrere Treffen, Besichtigungen und Termine beider universitärer Seiten gab. Im Juni 2017 fand zudem ein weiterführender Workshop zu vernetzten Themenbereichen in Greifswald statt.

ein Partner. Es handelt sich dabei um ein transatlantisches Projekt zum Aufbau eines gemeinsamen Harvesters\textsuperscript{3} für Belief legends\textsuperscript{4}. Vgl.: [17, 16].

Literatur

6. Heimatbund Mecklenburg: Zur Flurnamenforschung; Aufruf zur Flurnamenforschung; Fragebogen für die Flurnamenforschung. Zeitschrift des Heimatbundes Mecklenburg (Landesverwalt des Bundes Heimatschutz) 3(1), 16–22 (1908)

\textsuperscript{3} Zum Einsammeln Daten, die mittels OAI-PMH-Schnittstellen bereitgestellt werden.
\textsuperscript{4} dt. Glaubenslegenden
Reconstructing Graph Pattern Matches Using SPARQL

Stephan Mennicke\textsuperscript{1}, Denis Nagel\textsuperscript{2}, Jan-Christoph Kalo\textsuperscript{2}, Niklas Aumann\textsuperscript{2}, and Wolf-Tilo Balke\textsuperscript{2}

\textsuperscript{1} Institut für Programmierung und Reaktive Systeme, TU Braunschweig, Germany
mennicke@ips.cs.tu-bs.de
\textsuperscript{2} Institut für Informationssysteme, TU Braunschweig, Germany
denis.nagel,n.aumann}@tu-bs.de,\{kalo,balke}@ifis.cs.tu-bs.de

Abstract. Pattern matching is the foundation for handling complex queries to graph databases. Commonly used algorithms stem from the realm of graph isomorphism and simulations, being well understood theoretical frameworks. On the practical side, there are established graph query languages that often allow for a wide variety of query tasks, often even beyond pattern matching. However, very little is known how graph queries from common query languages relate to graph pattern matching relations. In this paper, we propose a study in this respect for SPARQL, the W3C recommendation for querying RDF data. The homomorphic nature of the SPARQL semantics allows for a straightforward formulation of graph-isomorphic matching. However, the somewhat artificial nature of these queries motivates the study of sole basic graph patterns, the foundational concept of SPARQL. For basic graph patterns, we show a correspondence to strong simulation, an efficient graph pattern matching relation appreciated for its polynomial bound matches. In consequence, graph query languages are capable of serving as generating frameworks for established graph pattern matching relations.

1 Introduction

Graph databases have gained lots of attention due to their popularity in emerging applications like the Semantic Web, social network analysis, or bio-technology. These graphs usually provide entity-centric data, in which nodes represent entities, while the edges model relations between entities. Several graph database query languages were developed, enabling users to query graph-structured data in an SQL-like fashion. Most notably, SPARQL is the W3C standard for querying Semantic Web data, and is also used for a wide range of applications [13]. On the foundational side of graph querying, graph pattern matching in terms of special homomorphisms forms the main influence. However, to the best of our knowledge, only little is known on the relationship between commonly used graph query languages — SPARQL in this paper — and other graph pattern matching relations that have been researched for decades in conceptual frameworks. Research in this area led to several complexity results, the development of
fast algorithms, and insights on the semantics of the respective relations, whose potential is, in our view, not yet fully exploited in the area of graph database querying. Lately, some effort were expended to use graph pattern matching algorithms for matching relations different from classical subgraph isomorphism, often appreciated due to their advantages in performance over traditional graph querying languages [9, 11].

In this paper, we study the relationship between graph pattern relations and graph query languages, here exemplified by SPARQL. Our first result is best explained by an example. Imagine a user writing a query to search for the directors and writers of movies that won an award, see Fig. 1(a). An isomorphic match to this query is, for instance, the movie Jurassic Park which won an Academy Award, was written by M. Crichton, and directed by S. Spielberg (cf. Fig. 1(b)).

The same result is achieved by the SPARQL query, depicted in Fig. 2. The first part, also called basic graph pattern, comprises variables for the graph pattern nodes and is arranged as triples representing the edges of the graph pattern. The filter condition at the end of the pattern ensures that each assignment to the variables is a bijective one, a necessary condition for graph-isomorphic matching. Being able to formulate such a query is not a coincidence. We prove, in Sect. 4, for every graph pattern there is a query returning every subgraph from a database that is isomorphic to the given pattern. A closer look at the filter condition raises the question, whether a user would be using such an artificial formulation. Removing the filter allows for (possibly unintended) variable assignments, and may produce answers as depicted in Fig. 1(c). Solely relying on basic graph patterns yields answers showing a (dual-)simulating character upon the original graph pattern, being the result of Sect. 5.

We provide partial and complete characterizations of graph pattern matching by SPARQL, based on basic graph patterns. We observe that dual simulation cannot be fully characterized, since the matching relation allows for arbitrary additions to matches, being also matches of the pattern. Strong simulation, an extension of dual simulation, removes this arbitrariness and is a matching relation renown for its efficient evaluation and its polynomial bound number of matches [9]. We find that SPARQL results may be used as building blocks to obtain all strong simulating matches. In return, strong simulation may also serve as a pruning method for SPARQL query engines. Sect. 3 provides basic notions.

---

Fig. 1: (a) An example graph pattern $P$ and (b) an isomorphic match of $P$ and (c) a (dual-)simulating match of $P$. 
SELECT *  
WHERE { ?v_{e_1} directed ?v_{e_3} . ?v_{e_2} wrote ?v_{e_3} . ?v_{e_3} awarded ?v_{e_4} .  
FILTER ( ?v_{e_1} != ?v_{e_3} && ?v_{e_1} != ?v_{e_3} && ?v_{e_1} != ?v_{e_4} && ?v_{e_2} != ?v_{e_3} && ?v_{e_2} != ?v_{e_4} && ?v_{e_2} != ?v_{e_4} ) } 

Fig. 2: The query for graph isomorphism of the graph pattern Fig. 1(a)

Related work and conclusions are given in Sect. 2 and Sect. 6. Due to space limitations, full proofs of the theorems are included in the appendix of this paper.

2 Related Work

Graph Pattern Matching is an extensively studied topic in various domains of computer science [5]. Its applications range from social network analysis, over structural analysis of chemical entities to various applications in the database domain, particularly in graph databases. Recently, emerging applications led to the trend of graph pattern matching relations, different from the canonical though costly candidate of graph isomorphism, with the goal of reducing structural requirements of the answer graphs. For example, the idea of simulation for graph pattern matching has been implemented for different graph database tasks [1, 3, 2]. Indeed, experiments have shown advantages of simulation-based matching relations when analyzing social network patterns, as they offer the possibility to collapse several nodes into one node and vice versa. Another recent case study in this respect are so called Exemplar Queries [11], representing an attempt to enable an easy access to databases without the need of knowing the formal requirements of a query language. Based on an example graph pattern from the database (the exemplar), the query process of Exemplar Queries checks for similarity, e.g., up to strong simulation, between the exemplar and other database structures, retrieving and ranking them for presentation to the user.

Graph Query Languages basically all are founded on the idea of graph pattern matching (with suitable substitutions) [14]. A matching mechanism common to most of these query languages is (sub-)graph isomorphism [4, 8]. Mainly due to the advances in the field of Semantic Web, SPARQL has become the W3C recommendation for querying Semantic Web data, i.e., RDF. A more detailed introduction to SPARQL follows in the next section. In general, graph query languages differ greatly with respect to their area of operation. Therefore, many different graph database operations, e.g., subgraph matching or adding new nodes, are considered, particularly when comparing the expressive power of different graph query languages [14]. Most languages solely rely on homomorphic, more specifically, isomorphic pattern matching, but their connection to other matching relations is not yet studied extensively. Therefore, here we give insights into this aspect of graph query languages with respect to SPARQL. The basic idea however, could also be applied to other graph querying languages, also
relying on the idea of basic graph patterns (e.g., Cypher or Gremlin). Regarding expressiveness of graph querying languages, in [7], the authors describe the graph query language GraphQL. It is based on a modified relational algebra that uses graph pattern matching for querying. They prove that their graph query algebra is relationally complete and therefore as expressive as relational algebra.

While we focus on graph isomorphism and simulations, the key question behind this work is not restricted to those relations. Many more comparison relations are discussed in the literature which may correlate very well with the semantics of graph query languages. For instance, the linear-time branching-time spectrum [6] provides several matching relations, under the term comparative semantics, used w.r.t. different aspects of system correctness. It contains comparative system relations for processes modeled as labeled transition systems, i.e., edge-labeled directed graphs with a distinct initial state. Most of the semantics come with a logical characterization in terms of a modal logic equipped with explicit quantification over edges to be traversed. Finding such a characterization is a common task, as it allows for expressing distinguishing characteristics of system behaviors in a precise manner. Similar to our subject, these characterizations express that two systems are equivalent whenever they satisfy the same logical formulas. In this paper, we try to adjust to the circumstances as imposed by SPARQL semantics.

Variants of graph homomorphism are also studied in the context of graph databases [4]. While the relations in our work always match an edge of a graph to other edges, as of preserving structure of a graph pattern to a certain extent, p-homomorphism takes each edge and maps it to paths in the match graph. This way, a p-homomorphic match graph may show a very different structure, being only loosely coupled with the graph pattern. Instead, p-homomorphic matching relies on a metric of node similarity.

3 Preliminaries

In this section, we define graphs, graph databases complemented by the general concept of graph pattern matching. Furthermore, we introduce an algebra of SPARQL.

A \((\Sigma\text{-})\)labeled directed graph is a triple \(G = (V, \Sigma, E)\), where \(V\) is a finite set of nodes, \(\Sigma\) a finite alphabet, and \(E \subseteq V \times \Sigma \times V\) a labeled edge relation. We represent an edge \((v, a, v') \in E\) by \(v \xrightarrow{a} v'\). Labeled directed graphs range over by \(G, G_1, G_2, P\) with node sets \(V, V_1, V_2, V_p\), a fixed alphabet \(\Sigma\) common to all graphs, and edge relations \(E, E_1, E_2, E_P\) with respective notations \(\xrightarrow{\rightarrow}, \xrightarrow{\rightarrow_1}, \xrightarrow{\rightarrow_2}, \xrightarrow{\rightarrow_p}\). A graph \(G_1\) is a subgraph of a graph \(G_2\), denoted \(G_1 \subseteq G_2\), iff \(V_1 \subseteq V_2\) and \(E_1 \subseteq E_2 \cap (V_1 \times \Sigma \times V_1)\). Two graphs \(G_1\) and \(G_2\) are isomorphic, written \(G_1 \cong G_2\), iff there is a bijective function \(\kappa : V_1 \rightarrow V_2\) such that \(v \xrightarrow{a} v'\) if and only if \(\kappa(v) \xrightarrow{a} \kappa(v')\). \(\kappa\) is called an isomorphism between \(G_1\) and \(G_2\). For two nodes \(v, v'\) of a graph \(G\), we define the distance \(\text{dist}_G(v, v')\) to be the length of the shortest undirected path from \(v\) to \(v'\). If there is no path between \(v\) and \(v'\), \(\text{dist}_G(v, v') = \infty\). However, we are interested in connected graphs throughout the
rest of the paper, i.e., for every two nodes \( v, v' \), \( \text{dist}_G(v, v') \neq \infty \). The diameter of a graph \( G \) with node set \( V \), denoted \( \text{dia}(G) \), is the greatest distance between nodes in this graph, i.e., \( \text{dia}(G) := \max \{ \text{dist}_G(v, v') \mid v, v' \in V \} \).

Graph databases store objects from a countable universe \( O \) together with attributes over the objects as either relations between objects or properties of objects. As an example, consider an \textit{isFriendOf} relation between objects referring to persons who are friends in social networks. Properties of objects are expressed as assignments of concrete data values, also called literals, from a usually infinite domain \( L \), to an object, e.g., the age of a person as a positive integer. Inspired by the treatment of literals in RDF, objects as well as literals are represented as nodes in a graph database. Relation symbols and property symbols stem from a finite set, here \( \Sigma \). A graph database is a directed labeled graph \( DB = (V, \Sigma, E) \) with a finite set of database objects \( V \subseteq O \cup L \).

A graph pattern is a connected graph \( P = (V_P, \Sigma, E_P) \). A subgraph \( G \) of a graph database \( DB \) is an isomorphic match of \( P \) (in \( DB \)) iff \( P \cong G \). By \( [P]_{DB}^\cong \) we denote the set of all isomorphic matches of \( P \).

Since SPARQL aims at querying RDF-stored data, the basic building blocks of the query language are triples of the form \((s, p, o)\). Subjects \((s)\) refer to objects in \( O \) or variables being assigned by actual database objects during the querying process. Objects \((o)\) may further be associated with literals from \( L \). Predicates \((p)\) are thought of as the relation and property symbols in \( \Sigma \) gluing together subjects with objects. Variables are place-holders for actual objects or literals as present in concrete databases. The result of a SPARQL query process is an assignment of objects and literals to the variables mentioned in a query expression. We denote the set of all variables by \( V \). Notation and semantics are based on [12].

Sets of such \((s, p, o)\)-triples are called basic graph patterns (BGP), which we will assume to be graphs. For every BGP \( B = \{(s_1, p_1, o_1), \ldots, (s_k, p_k, o_k)\} \) \((k \geq 0)\) where \( s_i \in O \cup V \), \( p_i \in \Sigma \), and \( o_i \in O \cup L \cup V \) \((i = 1, \ldots, k)\), the associated graph is \((V_B, \Sigma, B)\) such that \( V_B = \{s_1, \ldots, s_k, o_1, \ldots, o_k\} \). Notice that the nodes in the graph may also be variables. In fact, from the next section on, we employ BGPs in which all nodes are variables. In this paper, BGP \( B \) and its graph representation are used interchangeably. By \( \text{vars}(B) \) we denote the set of all variables occurring in \( B \).

The semantics of SPARQL BGPs \( B \), and henceforth of SPARQL queries \( Q \), is given in terms of assignments of objects and literals to variables in \( B \) \((Q, \) respectively). An assignment is a partial function \( \mu : V \rightarrow (O \cup L) \). By \( \mu(B) \) we reference the graph where each variable node \( v \in \text{vars}(B) \) is replaced by \( \mu(v) \). We define \( \text{dom}(\mu) := \{v \in V \mid \mu(v) \text{ is defined}\} \). An assignment \( \mu \) is valid w.r.t. \( B \) and a graph database \( DB \) iff (a) \( \text{dom}(\mu) = \text{vars}(B) \) and (b) \( \mu(B) \) is a subgraph of \( DB \). Thus, \( \mu \) is a graph homomorphism. The set of all valid assignments w.r.t. \( B \) and a graph database \( DB \) forms the foundation of the SPARQL query semantics. We denote this set by \( [B]_{DB}^\equiv \).

The second concept of SPARQL we use is that of filter conditions, also called built-in conditions. Filters are used to further restrict the set of (valid) assign-
ments of a SPARQL query. Thereby, we may check for equality (=) or inequality (<, ≤, ≥, >) of variable assignments, objects, and literals. The usual propositional connectives (\&\&, ||, ¬) are used to build complex constraints. For a full list of features we refer to the W3C recommendation report [13]. We denote by $\mu \models \varphi$ that assignment $\mu$ satisfies filter condition $\varphi$. Let $Q$ be any SPARQL query, e.g., $Q = B$ for a BGP $B$, and $\varphi$ a filter condition. Then $Q$ filter $\varphi$ is a SPARQL query. The semantics is given recursively in terms of the assignments from $\llbracket Q \rrbracket_{DB}$ such that every assignment conforms to $\varphi$. Thus, $\llbracket Q \text{ filter } \varphi \rrbracket_{DB} := \{ \mu \in \llbracket Q \rrbracket_{DB} \mid \mu \models \varphi \}$.

Throughout the paper, we make use of BGPs adjoint with filter conditions. The last SPARQL concept we need throughout Sect. 5 is that of a join of two queries. $Q_1$ and $Q_2$ represent the join of $Q_1$ and $Q_2$. Given two assignments $\mu_i \in \llbracket Q_i \rrbracket_{DB}$ ($i = 1, 2$). Then they are compatible if for every $v \in \text{dom}(\mu_1) \cap \text{dom}(\mu_2)$, $\mu_1(v) = \mu_2(v)$. Compatible assignments may be joined, thus, $\llbracket Q_1 \text{ and } Q_2 \rrbracket_{DB} := \{ \mu_1 \cup \mu_2 \mid \mu_i \in \llbracket Q_i \rrbracket_{DB} \text{ are compatible} \}$. The remaining operations of union and optional queries are not needed in this paper.

Next, we show that for every graph pattern $P$, there is a SPARQL query $Q_P$ that uses a BGP and a specific filter condition to obtain all graph-isomorphic matches of $P$ from a database $DB$. Please note that we assume graphs to be loop-free, throughout the paper, i.e., for each edge $v_1 \xrightarrow{a} v_2$, $v_1 \neq v_2$.

4 Querying like Graph Isomorphism

A graph pattern $P$ gives rise to a canonical BGP. In order to characterize isomorphic matches of $P$ from some database $DB$, we need to adapt the nodes of $P$ to obtain the possibility of arbitrary assignments of database objects/literals to the nodes of $P$. In SPARQL terms, this adaptation is performed by exchanging each node by a variable. Fig. 2 gives an example conversion in the where-clause, excluding the filter condition.

Definition 1. Let $P = (V_P, \Sigma, E_P)$ be a graph pattern. Define $\nu : V_P \rightarrow V$ such that $\nu(v) := v$. The BGP of $P$ is defined as the graph $\hat{P} := (V_P, \Sigma, \hat{E}_P)$ such that $V_P = \{ \nu(v) \mid v \in V_P \}$ and $(\nu(v), a, \nu(v')) \in \hat{E}_P$ iff $(v, a, v') \in E_P$.

From a graph-theoretic perspective, it directly follows that each graph pattern $P$ is isomorphic to its BGP $\hat{P}$ by isomorphism $\nu$. Every assignment $\mu \in \llbracket \hat{P} \rrbracket_{DB}$ is a homomorphism, but it is not guaranteed that each graph $\mu(\hat{P})$ is isomorphic to $P$. We enforce bijectivity of $\mu$ by adding a filter condition checking that for each two distinct nodes $v, v' \in V_P$, $\mu$ assigns different objects from the database.

Definition 2. Let $P$ be a graph pattern with set of nodes $V_P = \{ v_1, v_2, \ldots, v_n \}$. Define a filter condition for $P$ alongside $\nu$ (Def. 1) as $\varphi_P = \bigwedge_{i<j} \nu(v_i) \neq \nu(v_j)$.

The $P$-query for isomorphism is $Q_{=\Rightarrow}(P) := \hat{P} \text{ filter } \varphi_P$.

Reconsider the query given in Fig. 2 as an example query for isomorphism. We now show that the assignments of a $P$-query for isomorphism are equivalent to the set of all isomorphic matches w.r.t. $P$. 

157
Theorem 1. Let $DB$ be a graph database and $P$ a graph pattern. Then $G \in \llbracket P \rrbracket_{DB}$ if and only if there is an assignment $\mu \in \llbracket Q \rrbracket_{DB}$ such that $\mu(\hat{P}) = G$.

The proof exploits that assignments $\mu$ from respective queries already are isomorphisms. This means that by using BGP and a specific filter condition, we reach the same expressive power as isomorphic graph pattern matching. In a way, Theorem 1 states that SPARQL is complete w. r. t. graph-isomorphic matching. Graph isomorphism is among the strongest similarity-based matching relations between graphs. Reducing the constructed SPARQL query in this section to only the BGP yields a similar result between SPARQL and graph homomorphism, since a valid assignment amounts to a graph homomorphism. However, as we will see throughout the next section, BGP itself show an interesting correspondence to graph pattern matching by similarity, or more specifically, strong simulation. Strong simulation is renown for its efficient evaluation, compared to isomorphic matching, and its polynomial bound on the number of matches [9].

5 Basic Graph Patterns Query for Simulations

In this section, we first show that every valid assignment of a BGP $\hat{P}$ corresponds to a so-called dual simulating match of the underlying graph pattern $P$. Thereupon, we argue that there is no reasonable way to capture all dual simulating matches by a single SPARQL query, since a match may be extended arbitrarily, even by database relations that are not mentioned by pattern $P$. The notion of strong simulation extends dual simulation by restricting (1) the size of the matches by the diameter of the pattern and (2) all occurring nodes and edges to match nodes and edges in the pattern. We show this restriction to be sufficient to prove the existence of a SPARQL query that captures all strong simulating matches. We introduce the necessary notions as needed.
5.1 Dual Simulation

Simulations stem from studies of (concurrent) system behavior [10, 6]. Intuitively, system 2 simulates system 1 if whatever action system 1 performs, system 2 is capable of mimicking this behavior. When we look at graphs, the notion carries over in the sense that we assume the nodes of the graphs to play the role of states and the labels on edges to represent the actions. We continue the introductory example (cf. Fig. 1). Considering the graph pattern $P$ of Fig. 1(a), the graph in Fig. 3(a) is a simulating match of $P$. Starting by node $v_1$, $P$ may only perform the actions, as represented by labeled edges ‘directed’ and ‘awarded’, in this order. Alternatively, the graph may also perform the sequence ‘wrote’ and ‘awarded’ when starting in $v_2$ or just ‘awarded’ when starting in $v_3$. Identical actions in the same order may be performed in the match graph. Therefore, it is indeed a simulating match of $P$. Formally, a simulation is a binary relation over the nodes of the respective graphs such that each node of the simulated graph is actually simulated in the above-mentioned sense. A dual simulation extends the notion of simulation in such a way that it also looks at actions going backwards from a simulated node. While Fig. 3(a) represents a simulating match for $P$, it is not a dual-simulating match. This is because the pattern may go backwards from $v_4$ via ‘awarded’ and then face the choice to go for ‘directed’ or ‘wrote’, which is not possible in Fig. 3(a). Fig. 3(b), on the other hand, is a valid dual-simulating match. In the study of concurrent systems, dual simulation does not have a feasible interpretation, since we are usually not able to let a system revert its actions. For graphs, in general, and graph database objects, this makes sense, since an object may be part of a relation, either as subject or object, which is equally important w.r.t. the represented relation.

**Definition 3.** Let $G_i = (V_i, \Sigma, E_i)$ ($i = 1, 2$) be two graphs. A dual simulation between $G_1$ and $G_2$ is a relation $S \subseteq V_1 \times V_2$ such that (a) for each node $v_1 \in V_1$, there is $v_2 \in V_2$ such that $(v_1, v_2) \in S$ and (b) for each $(v_1, v_2) \in S$,

1. $v_1 \xrightarrow{a} v'_1$ implies that there is a $v'_2 \in V_2$ with $v_2 \xrightarrow{a} v'_2$ and $(v'_1, v'_2) \in S$,
2. $v'_1 \xrightarrow{a} v_1$ implies that there is a $v'_2 \in V_2$ with $v'_2 \xrightarrow{a} v_2$ and $(v'_1, v'_2) \in S$.

Let $DB$ be a graph database and $Q$ be a graph pattern. A subgraph $G$ of $DB$ is a dual simulating match of $Q$ in $DB$ iff $Q \preceq_D G$. The set of all dual simulating matches of $Q$ in $DB$ is denoted by $\llbracket Q \rrbracket_{DB}^D$.

It is easy to show that the union of two dual simulations again yields a dual simulation. Since an assignment to a BGP is a homomorphism, we obtain only matches respecting at least the edge structure of a pattern. Therefore, each assignment corresponds also to a dual simulating match.

**Proposition 1.** Let $DB$ be a graph database and $P$ a graph pattern. Then for all $\mu \in \llbracket P \rrbracket_{DB}$, it holds that $\mu(P) \in \llbracket P \rrbracket_{DB}^D$.

For every $\mu \in \llbracket P \rrbracket_{DB}$, $S_\mu = \{(v, \mu(v(v))) \mid v \in V_P\}$ is the desired dual simulation. The converse does not hold, in general. This is because given a dual simulating match of a graph pattern $P$, every graph that contains this match as a subgraph is also a match. In theory, if the size of the database as well as the
maximum in- and out-degrees of the database nodes are given, one could try to iteratively build BGPs extending the given graph pattern by structures allowed for dual simulation. Such a methodology is rather costly, since we may assume the database to be very large. In contrast, a query is often smaller, easily rendering the answers produced by the enlarged query useless. In fact, if there is at least one dual-simulating match in the graph database, then the graph database itself is also a match. Strong simulation overcomes these issues by limiting the size of matches by the diameter of the pattern. Matches to the graph pattern are locally bounded. Furthermore, irrelevant edges in the match are filtered out, letting a characterization of the respective SPARQL queries come in reach.

In order to exclude irrelevant edges, and thus, also nodes from a matching, the notion of match graph of a dual simulation $S$ is introduced. In a match graph of $S$, each node and each edge play a role in the simulation. Formally, a graph $G$ is a match graph w. r. t. dual simulation $S$ iff (a) for each node $v$ of $G$, there is a pair $(x,v) \in S$ for some node $x$ of the pattern, and (b) for each edge $v_1 \xrightarrow{a} v_2$, there is an edge $u_1 \xrightarrow{a} u_2$ in the pattern such that $(u_i, v_i) \in S$ ($i = 1, 2$). While strong simulation considers match graphs of special dual simulations, as we will explain in the next subsection, also dual simulation may benefit from this notion. By $|G|$ we denote the size of $G$, defined as the number of nodes in $G$.

**Lemma 1.** Let $DB$ be a graph database and $P$ a graph pattern. For all matches $G \in [[P]]_{DB}^S$ with dual simulation $S$ such that $|G| \leq |P|$ and $G$ is a match graph w. r. t. $S$, there is a $\mu \in [[\hat{P}]]_{DB}$ such that $\mu(\hat{P}) = G$. 

**Proof.** Let $G$ be a dual simulating match of $P$ under dual simulation $S$, as required. Then $S$ is a homomorphism between $P$ and $G$. Each node of the pattern has exactly one simulating node in $G$, since $G$ is a match graph w. r. t. $S$ (cond. (a)) and $|G| \leq |P|$. As $G$ dual simulates $P$, an edge in $P$ is reflected by an edge in $G$, homomorphically. From match graph cond. (b), it follows that each edge in $G$ is reflected by some edge in the pattern. By taking $\mu = S$, we get $\mu(\hat{P}) = G$. 

5.2 Strong Simulating Matches

The last Lemma has given an exact characterization of valid assignments of SPARQL BGPs in terms of matching by dual simulation. In this section, we look at further restrictions on dual simulations, culminating to the notion of strong simulation. Strong simulation is an extension of dual simulation, i.e., a match still needs to dual simulate the pattern (e.g., reconsider Fig. 3(a) as not strong simulating), but this time, a simulation $S$ certifying for dual simulation must be maximal, i.e., any other dual simulation is already contained in $S$. Uniqueness of the maximal dual simulation is easy to show and may be found, e.g., in [9]. Furthermore, a prospective graph needs to be a match graph w. r. t. maximal dual simulation $S$. 

160
Strong simulation aims at keeping possible matches locally constrained such that the size of a match is bounded and also the overall number of matches is non-exponential. In order to meet this locality requirement, matches are bounded in the diameter $d$ of the pattern in such a way that for every match, there is a node having distance at most $d$ to any other node, e.g., Fig. 3(b) is not strong simulating to the graph pattern in Fig. 1(a), because the match graph is disconnected. Therefore, the locality requirement is violated. Formally, we require a match to be a subgraph of a so-called ball of the database $DB$. Let $v$ be a node of $DB$ and $r \in \mathbb{N}$ a radius. Then the subgraph of $DB$ containing node $v$ and all nodes and edges reached from $v$ in at most $r$ steps (backwards and forwards along the edges) is called a ball of $DB$, denoted $\hat{DB}[v,r]$. For strong simulation, the radius is chosen to be the diameter of the graph pattern.

**Definition 4.** Let $DB$ be a graph database and $P$ a graph pattern ($\text{dia}(P) = d$). Subgraph $G$ of $DB$ is a strong simulating match of $P$ in $DB$ iff there is a node $v$ such that $G$ is a subgraph of $\hat{DB}[v,d]$ containing $v$ with the following properties: (1) $P \preceq G$ by maximal dual simulation $S$ and (2) $G$ is the match graph of $S$. By $\{P\}'_{\hat{DB}}$ we denote the set of all strong simulating matches of $P$.

Both, Fig. 1(b) and Fig. 1(c) are strong simulating matches of the pattern in Fig. 1(a). Ma et al. [9] call a strong simulating match perfect subgraph. In the spirit of Proposition 1, one can show that each assignment also corresponds to a strong simulating match. Again, not all matches can be recovered by a simple transformation of a graph pattern into a BGP. However, since the size of the matches is bounded by the diameter of the given graph pattern, an iterative method, like the one explained at the end of Sect. 5.1, may be feasible.

The rest of this section is devoted to proving the existence of a SPARQL query for graph pattern $P$, i.e., a $P$-query for strong simulation. Therefore, we first look at the SPARQL answers, already giving us strong simulating matches, and join them to bigger answers, similar to the SPARQL AND-operation (cf. Sect. 3). We then prove that for every strong simulating match there is a set of SPARQL assignments that amounts to the graph by joining the components. The resemblance of our join operator allows us to conclude that there is a SPARQL query reflecting the strong simulating matches of a graph pattern.

The join of two subgraphs of $DB$ is the union of both graphs if they are compatible. Two graphs are compatible iff they share at least one node. Remember that we assume only connected graphs as graph patterns, justifying this stronger condition compared to the AND-operator of SPARQL.

**Definition 5.** Let $DB$ be a graph database. Two subgraphs $G_1, G_2 \subseteq DB$ are compatible iff $V_1 \cap V_2 \neq \emptyset$. The join of compatible graphs $G_1$ and $G_2$ is defined as the graph $G_1 \join G_2 = (V_1 \cup V_2, \Sigma, E_1 \cup E_2)$.

Joining arbitrary strong simulating matches still yields dual simulating matches, but the locality requirement may be violated. Therefore, a second form of compatibility is necessary which restricts the possible combinations of strong simulating matches to be joined. For a graph $G$ and radius $r \in \mathbb{N}$, $\mathcal{C}(G,r)$ denotes the
set of center nodes of $G$ w.r.t. $r$, i.e., all other nodes may be reached in at most $r$ steps (ignoring the directions of the edges). Restricting $\gg\gg$ to only join center nodes w.r.t. the diameter of the pattern maintains the locality requirement of strong simulation.

**Definition 6.** Let $DB$ be a graph database and $r \in \mathbb{N}$ a radius. Compatible subgraphs $G_1, G_2 \subseteq DB$ are $r$-compatible iff $V_1 \cap V_2 \subseteq C(G_1, r) \cap C(G_2, r)$.

While $\gg\gg$ alone is commutative and associative, we lose associativity requiring $r$-compatibility. Therefore, we assume left-associativity of $\gg\gg$ throughout the rest of the paper. The join of two strong simulating matches of a graph pattern $P$ is again a strong simulating match of $P$ if the matches are $d$-compatible with $d = \text{dia}(P)$. This is because the union of two (maximal) dual simulations yields a (maximal) dual simulation and, as long as only center nodes are joined, the distance requirement of strong simulation remain unaffected. This insight paves the way for the main theorem of this section. Every strong simulating match of pattern $P$ may be reconstructed out of assignments of query $\hat{P}$. Since empty matches are trivially constructible, from zero BGP assignments, we rule out this case in the next theorem.

**Theorem 2.** Let $DB$ be a graph database, $P$ a graph pattern with diameter $d$, and $G \in \{P\}_{DB}^{ \gg\gg }$ be non-empty. There are assignments $\mu_1, \mu_2, \ldots, \mu_k \in \{\hat{P}\}_{DB}$ such that for each $0 < j < k$, $\mu_1(\hat{P}) \gg\gg \ldots \gg\gg \mu_j(\hat{P})$ and $\mu_{j+1}(\hat{P})$ are $d$-compatible and $\mu_1(\hat{P}) \gg\gg \mu_2(\hat{P}) \gg\gg \ldots \gg\gg \mu_k(\hat{P}) = G$.

**Corollary 1.** Let $DB$ be a graph database and $P$ a graph pattern. Then there exists a $P$-query for strong simulation $Q_{\gg\gg}(P)$.

**6 Conclusion**

We provided novel insights into the relation between the widespread graph query languages, exemplified by SPARQL, and graph pattern matching. Thereby, we obtained a fresh look at the expressive power of graph query languages w.r.t. well-understood graph pattern matching relations. To the best of our knowledge, this is the first attempt that characterizes graph pattern matching relations by state-of-the-art graph query languages. Our findings are general in that every query language complete w.r.t. the SPARQL semantics we used (cf. Sect. 3), may reproduce our theorems. By the homomorphic nature of the SPARQL semantics, it was possible to formulate simple queries consisting only of a BGP and a filter condition that constructs all isomorphic matches of a given graph pattern (cf. Theorem 1). Removing the somewhat artificial filter condition from $P$-queries for graph isomorphism, thus only considering BGP s, yielded queries returning dual simulating matches. From a computational point of view, the construction of a query capturing all dual simulating matches is rather costly. However, limiting matches to relevant (w.r.t. the pattern) nodes and edges allows to fully characterize dual simulation in this spirit (cf. Lemma 1). Ultimately, we showed the existence of SPARQL queries for matching by strong simulation.
an extension of dual simulation that locally restricts the size of the matches (cf. Theorem 2 and Corollary 1). As our first point for future work, we would like to give a constructive proof of Theorem 2 in order to make our findings applicable.

From a practical perspective, our results may be beneficial to query processing performance in very large graph databases. It is known that the evaluation of SPARQL queries itself is coNP-complete [12]. In contrast, graph pattern matching based on strong simulation only needs cubic time [9]. Query processing heuristics built on existing strong simulation algorithms could therefore lead to improvements with regard to general graph database query processing. Thereby, strong simulating matches may serve as over-approximations of BGPs, possibly reducing the number of relevant candidate assignments. Whether or not preprocessing of graph database queries by strong simulation leads to a significant reduction of computational time is left to an empirical evaluation. Further investigations on the interplay of BGPs and other graph query operations need to be performed.

References

3. Fan, W., Li, J., Ma, S., Tang, N., Wu, Y., Wu, Y.: Graph pattern matching: From intractable to polynomial time. PVLDB Endow. 3(1-2), 264–275 (Sep 2010)
8. Lee, J., Han, W.S., Kasperovics, R., Lee, J.H.: An In-depth Comparison of Subgraph Isomorphism Algorithms in Graph Databases. PVLDB Endow. 6(2), 133–144 (Dec 2012)
A Proof of Theorem 1

We show the two directions, separately.

if: Let $\mu \in \llbracket Q_\pi(P) \rrbracket_{DB}$ be an assignment. We need to show that $\mu(\hat{P})$ is an isomorphic match of $P$, i.e., $P \cong \mu(\hat{P})$. Therefore, we prove that $\mu \circ \nu$ is an isomorphism. By our observation that $\nu$ is an isomorphism and isomorphisms are preserved by function composition, it is sufficient to show that $\mu$ is an isomorphism. $\mu$ is injective due to the filter condition $\varphi_P$. $\mu$ is surjective by the definition of a valid assignment. Thus, $\mu$ is an isomorphism.

only if: Let $G$ be an isomorphic match of $P$. Thus, there is an isomorphism $\kappa$ between $P$ and $G$. Isomorphisms are closed under reversal, i.e., $\kappa^{-1}$ is also an isomorphism. Furthermore, $\nu^{-1}$ is an isomorphism for the same reason. We construct an assignment by composing these two isomorphisms as $\mu = \nu^{-1} \circ \kappa^{-1}$. $\mu$ is an isomorphism, thus a bijective homomorphism and a valid assignment for $\hat{P}$, resulting in $\mu(\hat{P}) = G$. $\Box$

B Proof of Theorem 2

We proceed by induction on the size of the graph $G$, estimated by the number $k$ of assignments needed to construct $G$. In the base case, $k = 1$, we look at graphs $G \in \llbracket P \rrbracket_{DB}$ with $|G| \leq |P|$. The statement follows directly from Lemma 1.

Assume for some $i$, the claim holds in case $k = i$ and any smaller number. We need to show that the claim also holds in case $k = i + 1$. The induction hypothesis implies that for every match $G$ with $|G| \leq i \cdot (|P| - 1) + 1$, there are at most $i$ assignments constructing the match. This follows from the base case constructing graphs of size at most $|P|$, while any further addition contributes at most $|P| - 1$ nodes to the graph, since by assumption at least one center node is involved in the join.

For case $k = i + 1$, we establish the following bound of the size of $G$,

$$|G| \leq (i + 1) \cdot (|P| - 1) + 1 = i \cdot |P| - i + |P| = \underbrace{i \cdot (|P| - 1) + 1}_{\text{induction hypothesis}} + (|P| - 1).$$

Towards a contradiction, suppose $G$ is not constructible from the assignments of the BGP $\hat{P}$. Then there is a largest constructible subgraph $G' \subseteq G$, which needs at most $i$ assignments, and they exist by the induction hypothesis. The subgraph of $G$ that has no shared nodes with $G'$ is bounded by $|P| - 1$ due to the above equation. There is at least one edge $v' \xrightarrow{\alpha} v$ (or $v \xrightarrow{\alpha} v'$, resp.) with $v'$ is in $G'$ and $v$ is in $G$ but not in $G'$. Since $G$ is a match of $P$, there is a smallest subgraph $G''$ of the database with $G'' \subseteq G$, containing $v' \xrightarrow{\alpha} v$ (or $v \xrightarrow{\alpha} v'$, resp.) and dual simulating $P$. If $G''$ is not a sugraph of $G$, then there is an edge outside of $G$ necessary to dual simulate pattern $P$; establishing a contradiction to the assumption that $G$ is a strong simulating match of $P$. Since the size of $G''$ is also bounded due to the equation above, $G''$ adds no more than $|P| - 1$ nodes to $G'$. By Lemma 1, there is an assignment of $\hat{P}$ that amounts to $G''$. Since this step may be repeated until no more edges remain, constructibility of $G$ is implied. $\Box$
Konzepte für das Forschungsdatenmanagement an der Universität Rostock – Extended Abstract

Concepts for the Management of Research Data at the University of Rostock (Extended Abstract)

Ilvio Bruder, Andreas Heuer, Sebastian Schick und Sascha Spors
Universität Rostock

Abstract: Research Data Management aims at gathering, capturing, storing, tracking, and archiving all the data being produced in scientific projects and experiments. Besides these data, all the processing steps on these data - eventually resulting in scientific publications - have to be stored as well.

Research Data Management is not only a scientific discipline in Computer Science. Universities and Research Institutes have to provide organizational structures and processes and pragmatic solutions (hardware and software resources) to implement first, simple tasks of Research Data Management.

In this paper, we sketch the organizational, pragmatic, and research aspects of Research Data Management from a local (University of Rostock) point of view. At the University of Rostock, we have wider experiences with research data management in marine biology and medical research. The research aspects are part of modern database research topics such as temporal databases, data integration, schema evolution, and provenance management.

1 Einleitung


Darüberhinaus ist Forschungsdatenmanagement ein aktuelles Forschungsthema, bei dem insbesondere moderne Datenbanktechnologien benötigt werden. In diesem Beitrag werden wir einige Forschungsansätze skizzieren, mit denen wir

An der Universität Rostock haben wir speziell Erfahrungen im Bereich von Informatik- und Elektrotechnik-Forschungsgebieten und im Bereich meeresbiologischer Forschungsdaten, mittlerweile aber auch verstärkt in anderen naturwissenschaftlichen und medizinischen Forschungsbereichen. Ein ganz anderer Bereich sind die weniger (mess- und sensor-)datengetriebenen geisteswissenschaftlichen Forschungsgebiete (Digital Humanities), in denen sehr dokumentzentriert gearbeitet wird. Im Folgenden werden wir uns eher mit den datengetriebenen Szenarien befassen.

2 Herausforderungen des Forschungsdatenmanagements

Es gibt im Forschungsdatenmanagement eine Reihe von Herausforderungen, die zum einen eher organisatorisch bzw. praktisch orientiert sind. Zum anderen gibt es auch interessante Herausforderungen in der Grundlagenforschung, gerade in Bezug auf Datenbanktechnologien. Die Herausforderungen in der Datenbank-Grundlagenforschung sind

- die Heterogenität der Daten,
- ein nicht oder nur unvollständig vorhandenes Schema,
- die Sicherung der Provenance (Herkunft) der Forschungsergebnisse sowie die Reproduzierbarkeit der wissenschaftlichen Auswertungen,
- die Spezifikation und Nachverfolgung von wissenschaftlichen Arbeitsabläufen (Scientific Workflows), sowohl organisatorisch als auch datentechnisch,
- die Einbettung und Speicherung von anwendungspezifischen Funktionen und Methoden, insbesondere zur Analyse von Daten,
- temporale Aspekte zur Reproduzierbarkeit von Auswertungen über Messdaten, die als Stromdaten ständig produziert werden
- sowie die Komplexität in den Auswertungen und Veränderungen in den Auswertungsroutinen über einen längeren Zeitraum hinweg.

Eher praktische Herausforderungen, die zeitnah mit pragmatischen und bereits vorhandenen Hardware- und Software-Lösungen implementiert werden müssen, sind

- Open Science, inklusive Zugriff auf und Präsentation von Daten und Ergebnissen,
- die Umsetzbarkeit von Lösungen in der Praxis, etwa durch flexible Architekturen,
- die Nachhaltigkeit der implementierten Lösungen,
- Usability bzw. Ergonomie des Softwaresystems
- sowie Lizenz- und Rechtsfragen bei benutzten Originaldaten und Softwarewerkzeugen zur Auswertung und Darstellung der Daten.
Andere Ansätze unterstützen auch kollaboratives Arbeiten und stellen die Erfassung, Verwaltung und Nutzung von Forschungs-Metadaten in den Vordergrund (wie LabBook: [11]). Weiterhin können Forschungsdaten und (datenbankgestützte) Auswertungen auch in das zu publizierende Dokument integriert werden wie in Janiform mit den Portable Database Files (PDBF) [5].

Einige der oben genannten Herausforderungen werden wir im Folgenden genauer diskutieren.

3 Forschungsdaten-Lifecycle


Grundlegendes Prinzip, das im Rahmen dieser wissenschaftlichen Arbeitsabläufe realisiert werden muss, ist das FAIR-Prinzip aus [20]. FAIR beinhaltet,
dass Daten auffindbar (findable), zugreifbar (accessible), interoperabel und wiederverwendbar (reusable) sein müssen. Wir werden nun zunächst einige organisatorische Maßnahmen sowie pragmatische Lösungen beschreiben, die dieses Prinzip fördern sollen.

4 Forschungsdatenmanagement an der Universität Rostock

An der Universität Rostock wurden und werden verschiedene organisatorische Konzepte sowie grundlegende als auch praxisnahe Lösungen entwickelt. Einige sollen im Folgenden kurz vorgestellt werden.

Rostocker Modell


Forschungsdatenworkflow und Publikationsworkflow an der Universität Rostock


Open Science und pragmatische Lösungen für die kurzfristige Umsetzung des Forschungsdatenmanagements


Zur Open Science gehören eine Reihe von Begrifflichkeiten und Konzepten, um die Wissenschaft transparent und nachvollziehbar für jedermann anzu bieten (siehe auch [18]):

- Open Source – freie Verfügbarkeit des Sourcecodes und der im Forschungsprojekt verwendeten Werkzeuge
- Open (Science) Data – freie Verfügbarkeit der wissenschaftlichen Daten. Dazu gehören auch Protokolle, Beschreibungen, Kalibrierdaten, etc.
- Open Access – freier Zugriff auf die veröffentlichten Ergebnisse, wie Forschungsartikel
- Open Methodology – Beschreibung aller verwendeten Methoden, die für das Forschungsergebnis genutzt und entwickelt wurden
- Open Notebook Science – freie Verfügbarkeit sämtlicher Basisdaten, Aufzeichnungen, Planungs- und Einsatzinformationen zu einem Forschungsprojekt

Gerade die angesprochenen Probleme bei der Versionsverfolgung bei Forschungsergebnissen, die über einen längeren Zeitraum, etwa auch permanent, gewonnen und ausgewertet werden führten zu einigen grundlegenden Forschungsarbeiten, die derzeit in der Datenbank-Forschungsgruppe der Universität Rostock vorangetrieben werden.

5 Forschungsfragestellungen im Forschungsdatenmanagement


In der Systematik von Abbildung 1 werden wir uns dabei um die Auswertung von Primär- und Sekundärforschungsdaten in langfristigen Messreihen kümmern. Dabei ist die Reproduzierbarkeit dieser Auswertungsergebnisse für die Nachprüfbarkeit von Publikationsergebnissen in naturwissenschaftlicher Forschung wichtig. Das Provenance Management wird aber auch benötigt, um bei der Erforschung und Entwicklung von smarten Systemen (Assistenzsystemen) beurteilen zu können, warum die Situations- und Aktivitätserkennung in Assistenzsystemen in bestimmten Fällen fehlerhafte Ergebnisse geliefert hat. Sowohl bei langfristigen Messreihen als auch bei der Entwicklung von smarten Systemen sind daneben auch noch temporale Aspekte wichtig, da über die Zeit sowohl erfasste Daten als auch Auswertungsmethoden sich verändern können.
Provenance Management und Reproduzierbarkeit von Forschungsergebnissen

Bei der Reproduzierbarkeit von Forschungsergebnissen gibt es verschiedene Stufen. Man kann testen

- ob ein Ergebnis plausibel ist: hier ist die Fragestellung, ob eine Aussage in einer Publikation zu den gespeicherten Forschungsdaten passt; üblicherweise eine manuell durch Gutachter durchzuführende Tätigkeit, die u.a. durch Textanalysen softwaretechnisch zumindest unterstützt werden kann
- ob das Ergebnis nachvollziehbar ist: hier ist die Fragestellung, ob ein etwa tabellarisch aufbereitetes Ergebnis aus strukturierten Daten zu den gespeicherten Forschungsdaten passt; hier greifen schon weitere Analyseverfahren, die auf strukturierte Daten angewendet werden können
- oder ob das Ergebnis reproduzierbar ist: hier muss dasselbe Ergebnis aus strukturierten Daten auf Basis der gespeicherten Primärforschungsdaten mit der gleichen Methodik (etwa einer gespeicherten Analysefunktion) softwaretechnisch ohne Medienbruch wieder errechnet werden können.

Wenn man davon ausgeht, dass die Forschungsarbeiten auf Basis von Open Science durchgeführt werden, sollten die Forschungsergebnisse prinzipiell immer von den Primärforschungsdaten her reproduzierbar sein. Der entgegengesetzte Weg, vom Ergebnis zurück zu den Primärforschungsdaten, ist die Rückverfolgbarkeit von Forschungsergebnissen. Das zugrundeliegende Forschungsgebiet im Datenbankbereich ist das Provenance Management.

In der Provenance-Theorie unterscheidet man Where-, Why- und How-Provenance [16,4]. Die Fragestellungen dahinter sind:

- Where – woher kommen die Daten, die zu dem Ergebnis führten? Ergebnis ist die Herkunft der Daten in Form von zugrundeliegenden Datenbanken, Dateiverzeichnissen, Datensammlungen oder Repositorien.
- Why – welche Daten spielten exakt eine Rolle? Ergebnis sind die Einzel Daten (relational: die Tupel in Relationen), die in die Methodik eingeflossen sind.

Dabei sind Where- und Why-Anfragen auf die Forschungsdaten mit aktuellen Mitteln, wie z.B. einer Datenbankerweiterung der Open-Source-Datenbank PostgreSQL [7], machbar. Leider eignen sich die derzeit vorhandenen Provenance-Techniken nur unter sehr speziellen Randbedingungen, die üblicherweise bei Forschungsprojekten mit komplexen Auswertungsfunktionen nicht gegeben sind:

Die Auswertungen auf den Daten werden auf einer fixierten Datenbank vorgenommen: die Forschungsergebnisse berücksichtigen keine Updates (bei Stromdaten: ständig sich ändernde Menge an Primärforschungsdaten) auf den Forschungsdaten. Hier müssen wir die Provenance-Techniken mit temporalen Aspekten (siehe unten) verknüpfen.


**Temporale Aspekte im Forschungsdatenmanagement**

Für den Umgang mit Forschungsdaten sind Auswertungen von historischen Daten als auch die spätere Nachvollziehbarkeit von Forschungsergebnissen ein wesentliches Qualitätsmerkmal für eine Nachhaltigkeit dieser Forschung. Temporale Informationen über die Daten und Experimente sind nötig, um eine solche Nachhaltigkeit zu erreichen.


Integration statt Migration – Integrationspipeline

Bei der Organisation des Forschungsdatenmanagements werden meist Richtlinien (gibt es bei der DFG, BMBF, Hochschulen und Forschungseinrichtungen) oder Handbücher (ein allgemeines auch von der DFG anerkanntes ist [3]) ausgearbeitet, die die Vorgehensweise vorgeben. Aus bisherigen Erfahrungen wird allerdings eine solche Vorgabe nicht unbedingt beachtet und die Durchsetzung gestaltet sich ebenfalls oftmals schwierig. Kernpunkt eines nutzerfreundlichen Forschungsdatenmanagements sollte sein

- den Wissenschaftlern einen Mehrwert mit der Forschungsdatenverwaltung zu bieten,
- ihre Vorgehensweisen und Werkzeuge nicht zu ersetzen, sondern zu integrieren, sowie
- Services anzubieten, neben Beratung auch direkte, technische Unterstützung bei der Organisation und Umsetzung einer konkreten individuellen als auch verknüpften Forschungsdatenverwaltung.

Wissenschaftler stellen sich nicht zwangsweise komplett auf eine andere Art der Datenverwaltung um. Es ist bzgl. der Compliance besser, die speziellen Vorgänge der einzelnen Wissenschaftler in eine Gesamtstrategie zum Forschungsdatenmanagement zu integrieren.

Um die Forschungsdaten aus verschiedenen wissenschaftlichen Projekten zu integrieren sind Transformations- und Evolutionsprozesse der Daten, der Schemata und der Funktionalität vonnöten. Diese Vorgänge ähneln dem ETL-Prozess in Data Warehouses. Wir haben dazu eine Integrationspipeline entwickelt, die über verschiedene Stufen halbautomatisch Schemata extrahieren und integrieren kann [2].

6 Zusammenfassung

In diesem Beitrag wurden drei Aspekte des Forschungsdatenmanagements diskutiert. Es wurden die derzeitigen und geplanten organisatorischen Rahmenbedingungen an der Universität Rostock vorgestellt. Es wurden zunächst pragmatische Lösungen aufgezeigt, um zeitnah Lösungen für die Open-Science-Anforderungen umsetzen zu können. Schließlich wurden Forschungserfordernisse insbesondere aus dem Bereich der Datenbanktechnologie präsentiert.

Die Autorenguppe arbeitet dabei an verschiedenen Teilaspekten, etwa an den organisatorischen Rahmenbedingungen (Schick; Universitätsbibliothek Rostock, verantwortlich für Digitale Bibliotheken und Forschungsdaten), den pragmatischen Lösungen für die Open Science im Kontext eines startenden Sonderforschungsbereiches (Spors; Institut für Nachrichtentechnik) und an den genannten Forschungsfragenstellungen (Bruder, Heuer; Institut für Informatik, Forschungsgruppe Datenbanken).
Literatur


Bringing computation to the data(base) with Core Data Services (CDS)

Stefan Bäuerle, Alexander Böhm

SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
stefan.baueuerle@sap.com, alexander.boehm@sap.com

Bringing computations close to the data source is one of the key concepts of high-performance database management systems. This specifically holds true for state-of-the-art systems like HyPer, Microsoft SQL Server, or SAP HANA that are making efficient use of modern hardware by exploiting in-memory storage, vector instructions, and sophisticated algorithms for query optimization and execution. To benefit from the capabilities of these systems, it becomes increasingly important for business applications to efficiently push their data-intensive parts to the DBMS.

Capturing and communicating the computational intent from the application to the DBMS is often not trivial, as the DBMS is either abstracted away using object-relational mapper interfaces like Hibernate, or as the language mismatch between imperative, object-oriented languages like Java and the declarative, SQL-based query processing makes it difficult for developers to express non-trivial database operations.

In this presentation, we discuss how some of these issues can be tackled using the Core Data Services (CDS) framework that is both supported by SAP’s HANA database and the Netweaver application server stack. By combining concepts from declarative query languages like SQL and XPath with functionality developers are familiar with from object-oriented languages or even aspect-oriented programming (i.e. the “.”-operator for member access and the concept of annotations), CDS enables to express complex business logic with only a few lines of code – where equivalent SQL statements span multiple pages instead (see Figure 1 below for an example).

--CDS
SELECT FROM BSEG {bkpf.mandt, bukrs.butxt, mara.matxt,
    SUM(menge)) AS menge2 }
WHERE bkpf.txkrs <> 0 AND menge > 0
GROUP BY bkpf.mandt, bukrs.butxt, mara.matxt;

--SQL
SELECT BKPF.MANDT, T001.BUTXT, MARA.MATXT,
    SUM(BSEG.MENGE)
FROM BKPF JOIN BSEG ON BKPF.MANDT = BSEG.MANDT
    AND BKPF.BUKRS = BSEG.BUKRS AND BKPF.BELNR = BSEG.BELNR
    AND BKPF.GJAHR = BSEG.GJAHR
    JOIN MARA ON BSEG.MANDT = MARA.MANDT
    AND BSEG.MATNR = MARA.MATNR
    JOIN T001 ON BSEG.MANDT = T001.MANDT
    AND BSEG.BUKRS = T001.BUKRS
WHERE BSEG.MENGE > 0 AND BKPF.TXKRS <> 0
GROUP BY BKPF.MANDT, T001.BUTXT, MARA.MATXT

Fig. 1 Queries for retrieving a list of all ordered materials per companies
in CDS notation (top) and plain SQL (bottom)
Enterprise Database – Forgotten, or not?

Jürgen Bittner

SQL Projekt AG, Franklinstraße 25A, 01069 Dresden, Deutschland
Juergen.bittner@sql-ag.de

Abstract. With the appearance of the Database Technology the target of an enterprise wide application was joined, respective the integrated management of all relevant data of the enterprise within one database, because on this way a maximal usefulness of the new technology seemed achievable. Several preconditions have been avoided this target’s achievement. The current self-evident utilization of the database technology is referring predominantly to individual areas or departments of the organizations. But the requirements to support processes and analytics exceeding isolated systems were advancing permanently. Beside the further development of the database technology this led to integration solutions of different levels. They are developed and operated often with large time and effort. The Data Warehouse approach oriented to data consolidation for the purpose of reporting and analytics, but without a possibility to improve the underlying operational processes. The integration technology supports database crossing processes, can speed up these and reduce their cost, but has only limited influence on data quality. The objectives of data quality in connection with the integration technology as they are considered in the Master Data Management get the data consolidation in the focus again, but now, if the realization is consequent, with the requirement to provide the operational systems with improved data. Could this become the Enterprise Database? For it the software industry would need a technology conversion and above all a mindset change.

Keywords: Enterprise Database, Data Warehouse, Data Quality, Master Data Management.

1 Die ursprüngliche Vision


Das in den 70er Jahren bevorzugte Streben nach Integrierter EDV hat in zahlreichen Beiträgen und Konzepten für IMIS (Integrated Management Information Systems) die Datenbank-Technologie allerdings nur wenig berücksichtigt.


Ein grundlegendes Konzept, dass die Zielstellung einer Enterprise-Datenbank wenigstens begünstigt hätte, ist die 3-Ebenen-Architektur nach ANSI/X3/SPARC [1].

![3-Ebenen-Architektur des Datenbanksystems](image)

**Bild 1.** ANSI/X3/SPARC 1975: 3-Ebenen-Architektur des Datenbanksystems

Die primären Zielaspekte dieser Architektur sind Datenunabhängigkeit und Unterstützung des Datenbank-Entwurfs durch Trennung der Inhaltsaspekte von den Realisierungsstrukturen einerseits und den Verwendungsstrukturen andererseits. Damit
ergibt sich gleichzeitig ein Rahmen für die Vereinbarkeit dezentraler Sichten mit dem Ziel der Enterprise-Datenbank [2][3], was möglicherweise für künftige Anforderungen wieder ins Blickfeld kommen könnte.

2 Enterprise Information erfordert Konsolidierung

Der Bedarf nach Reports und Analysen, die über die Inhalte einzelner Datenbanken hinausgehen, führte in den 90er Jahren zur Data-Warehouse-Konzeption und auch entsprechenden Produkten.

![Diagramm](image)

**Bild 2. Data-Warehouse-Architektur**

Innerhalb des ETL-Prozesses ist hierbei die syntaktische und semantische Konsolidierung der verschiedenen Quellen-Datenbanken erforderlich (vgl. z.B. [4]). Auf der Schema-Ebene ist das ein Entwurfsprozess, der ein normalisiertes Schema für die Konsolidierungs-Datenbank liefern muss sowie die Transformationsregeln für die Überführung der Daten aus den Quellen in die Konsolidierungs-Datenbank. Bei entsprechendem Umfang der einbezogenen Quellen ist damit strukturell eine Enterprise-Datenbank gegeben.

Das Entwerfen einer solchen als „sekundär“ zu bezeichnenden Enterprise-Datenbank profitiert von der Tatsache, dass mit den Quellsystemen bereits umfangreiches Wissen gesammelt wurde. Andererseits wirkt erschwerend, dass Divergenzen und Inkompatibilitäten in der Menge der unabhängig voneinander entstandenen Quellen erkannt und durch Transformationen bereinigt werden müssen.

Vergleichend mit der Idealvorstellung einer „echten“ oder „primären“ Enterprise-Datenbank bleibt festzustellen:
Die Konsolidierungs-Datenbank ist kontinuierlich oder periodisch mit den Änderungen der Quellsysteme zu synchronisieren. Das kann erhebliche Kosten für Prozessorlast und Speichervolumen bewirken.

Ein virtuelles Data-Warehouse, das auf die Materialisierung der Konsolidierungs-Datenbank und/oder der Bereitstellungs-Datenbank verzichtet, verlagert die Kosten in die Query-Laufzeit und wird tendenziell zur Sicherung der Performance noch höhere Kosten für Prozessorlast verursachen. Der Entwurfsaufwand für die Konsolidierung ist natürlich unverzichtbar.

Eine Konsolidierungs-Datenbank oder das vollständige Data-Warehouse auf in-memory-Basis kann die beste Performance liefern, die damit erhöhten Kosten könnten aber die Frage nach Kostendämpfung durch einen primären Enterprise-Ansatz motivieren.


3 Prozessqualität erfordert Integration

Die Realität der heterogenen Anwendung von Softwaresystemen, die nicht oder wenig koordiniert entwickelt wurden, hat sich immer stärker ausgeprägt und dauert zu einem großen Teil bis heute an.

Ein entsprechender Integrationsserver verfügt im allgemeinen über eine Menge geeigneter Adaptoren, die die Datenstrukturen der Softwaresysteme senden und empfangen können sowie über eine leistungsfähige Mapping- und Routing-Funktionalität. Damit ergibt sich die Möglichkeit, aktuelle Daten der führenden Systeme so schnell wie nötig allen anderen zu übergeben, die diese Daten benötigen. Bestimmte Definitionen ermöglichen auch die Steuerung von Geschäftsprozessen.

Insoweit könnte man annehmen, dass die Integrationstechnologie die Zielstellung und Notwendigkeit einer Enterprise-Datenbank weitgehend ersetzen kann, ohne dass ein Enterprise-Datenbank-Schema explizit existiert. Dazu sei aber die Einschätzung erlaubt, dass die heutige Anwendung der Integrationstechnologie in der Praxis oft noch nicht das Niveau einer systematischen Enterprise Integration erreicht hat, stattdessen werden schrittweise Schwerpunktaufgaben behandelt, die nur sehr offensichtliche Prozess-Verbesserungen sichern.

Zu bedenken sei aber auch die seit mehr als 10 Jahren von Gartner und anderen publizierte Feststellung, dass der Aufwandsanteil für Integration in der IT weit mehr als 30% beträgt und dass Integration mit steigender Tendenz ein kritischer Teil der Informationsversorgung ist [5]. Auch wenn die Anwendung professioneller produktbasierter Integrationslösungen den spezifischen Aufwand reduziert, die Integrationsdichte der Softwaresysteme eines Unternehmens aber absolut bedeutend zunehmen wird, kann man wohl annehmen, dass dieser Anteil nicht kleiner wird.

In dieser Hinsicht könnte auch die Ressourcenfrage bedeutsam werden.

4 Datenqualität erfordert Master Data Management

Typischerweise sind Objekte wie Geschäftspartner oder Artikel jeweils in mehreren Datenbanken eines Unternehmens enthalten. Ein Integrationssystem kann neue Daten vom jeweiligen Ursprungsworkflow gemäß geltender Regeln zu den Systemen transfe-
rieren, die diese Daten benötigen. Ein hohes Niveau der Integration der Daten eines Unternehmens durch Vermeidung der Mehrfacherfassung und schnelles Synchronisieren sichert aber nicht automatisch eine hohe Datenqualität.

Hierzu haben sich Anforderungen ausgeprägt wie z.B. Validierung, Standardisierung, Vervollständigung, Klassifizierung und Identifikation mit Doublettenerkennung und –behandlung [6].

Die Realisierung dieser Anforderungen erfordert eine Architektur, die seit einigen Jahren grundlegend zum Master Data Management (MDM) gehört.

**Bild 3. Grob-Architektur des Master Data Management**

**“Master data management (MDM) is the practice of defining and maintaining consistent definitions of business entities (e.g., customer or product) and data about them across multiple IT systems and possibly beyond the enterprise to partnering businesses. MDM gets its name from the master and/or reference data through which consensus-driven entity definitions are usually expressed. An MDM solution provides shared and governed access to the uniquely identified entities of master data assets, so those enterprise assets can be applied broadly and consistently across an organization.”** [7]

alle Datenänderungen transactional ausgeführt werden können, sondern Entscheidungs- und Freigabeprozesse einschließen, die Workflows benötigen.

Die MDM-Nutzung an sich und speziell auch die Workflow-Notwendigkeit bewirken, dass die beteiligten Softwaresysteme Anpassungen erfordern.


Bild 4. Wichtige Anforderungen an Master Data Management

Auffällig ist dabei auch der Trend zu einer wesentlich verbesserten Darstellung der Semantik, was die Einbeziehung der Daten, die nicht zu den Master Data gezählt werden in den MDM Hub nahelegt. Insoweit erfüllt der MDM Hub die Rolle einer Konsolidierungsdatenbank im Data-Warehouse. Als Konsequenz wird dem MDM neben den operationalen Benutzungsszenarien Transactional und Workflow nun auch Analytics zugeordnet. In Bild 4 sind die Anwendungsprogramme mit operationaler Funktionalität wie üblich in Verbindung mit ihren spezifischen Datenbanken dargestellt.

### Tabelle 1. Wirkungsweise des MDM

<table>
<thead>
<tr>
<th>Implementation Style</th>
<th>Verfahren</th>
<th>Wirkung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry</td>
<td>Zentraler Index verweist auf alle Vorkommen eines Objekts</td>
<td>Daten bleiben unverändert</td>
</tr>
<tr>
<td>Consolidation</td>
<td>MDM liefert eine Konsolidierungsdatenbank</td>
<td>Operationale Daten unverändert, vergleichbar mit Data-Warehouse</td>
</tr>
<tr>
<td>Coexistence</td>
<td>MDM Hub verwaltet Teil der Daten, produziert golden records und publiziert diese an operationale Systeme</td>
<td>Zugriff auf diese Daten im MDM Hub oder in den operationalen Systemen möglich, weiterhin Mehrfachspeicherung</td>
</tr>
<tr>
<td>Centralized</td>
<td>MDM Hub verwaltet alle Daten zentralisiert</td>
<td>Zugriff auf die Daten (transactional oder workflow) im MDM Hub, keine Mehrfachspeicherung</td>
</tr>
</tbody>
</table>


![Bild 5. Coexistence-Architektur im Wandel](image-url)


Wenn auch im MDM mit Coexistence Style die Mehrfachspeicherung der Daten im Prinzip noch nicht beseitigt, aber vielleicht reduziert werden kann, wäre es interessant, fallweise zu untersuchen, ob damit oder auch im MDM mit Centralized Style relevante Einsparungen an Ressourcen (Speicher und Integrationslast) erzielt werden können.

5 Software-Entwicklung und Enterprise-Datenbank

Natürlich werden sich die Data-Warehouse-Technologie, die Integration und das Master Data Management noch lange weiterentwickeln. Es ist aber offensichtlich, dass alle 3 Technologien mehr oder weniger das Fehlen der Enterprise-Datenbank ausgleichen müssen. Besonders deutlich wird das bei einer konsequenten MDM-Zielstellung.

Sollte die ursprüngliche Vision einer Enterprise-Datenbank auf dem Wege des Master Data Managements oder direkt jemals eine Chance bekommen, müsste sich die datenbank-orientierte Software-Herstellung mit folgenden zwei Zielen auseinandersetzen, wobei die Datenbank im allgemeinen nicht mehr Bestandteil eines einzelnen Softwareproduktes sein dürfte, sondern in einem eigenständigen Entwurfsprozess entsteht und dynamisch weiterzuentwickeln ist:


Die zwangsläufig notwendigen Transformationen der Daten bewirken natürlich einen neuen Aufwand, der aber durch den Wegfall eines großen Teils der Transformationen des Datenaustauschs ausgeglichen wird.


Die schrittweise Entstehung einer Enterprise-Datenbank ist heute mehr als in der Frühzeit der Datenbank-Technologie vorstellbar. Wesentliche Voraussetzungen haben
einen besseren Stand. Ob das zweifellos große und weiter wachsende Nutzenpotential ein ausreichender Antrieb ist, die Schwierigkeiten auf dem Wege dahin zu überwinden, wird die Zukunft zeigen.

Verweise

Adaptive Business Process Visualization for a Data and Constraint-Based Workflow Approach

Eric Rietzke¹, Ralph Bergmann² and Norbert Kuhn¹

¹ University of Applied Science Trier, Germany
² University of Trier, Germany

Abstract. This paper introduces a novel approach which unifies a data-centric and a constraint-based workflow principle. This unified approach offers a scalable flexibility during the process execution and supports the requirements of knowledge intensive business processes. By the integration of a knowledge-based system, process definition and execution relevant data coincide on an ontology-based semantic net. The data, mainly driving the process, can be delivered by different sources or can be the result of an inference step by the underlying ontology. In such a case, AI technology plays an active role during the execution of processes and results in a division of labor with human actors. Such an AI contribution in the process execution must be presented explainable to the user and for a common understanding, this paper presents a concept for a business process visualization adapted to the introduced unified approach. Established strategies for the adaptation of process views are under examination and new strategies will be presented to utilize the integrated knowledge-based system for a semantic oriented process visualization.

1 Introduction

In today’s business, Process-Aware Information Systems [1] play an essential role for many companies. Often, these companies have to manage a large number of processes, involving different organizational units, a large number of human actors, and a multitude of activities. On the management and controlling level, users want to be informed about status and progress of such processes with a different granularity depending on their departmental focus. On the contrary, the participating user must execute certain process steps, where each of which requires that specific data or knowledge is available.

In many applications, a process is presented more or less in the same way as it was designed by the process designer. This usually does not fit to the specific needs of a user for the execution of an activity. Previous research has addressed these aspects [3, 9, 16] and have developed different concepts and approaches for business process visualizations (BPV).

Current BPV approaches are developed for an activity-centric (usually imperative) workflow principle where the activities are directly related to each other and form a control-flow or constraints define some rules for their execution. However, with view to the demand of knowledge intensive processes for
flexibility at design- and run-time [7], new concepts based on data-centric approaches are subject of investigation and have formed an active field of research [14, 2, 4] over the last decade. These approaches have in common that the activities are no longer directly related to each other. Instead, the activities are bound to the data elements which are required to perform an activity (input) or which are the result of it (output).

Such data-centric approaches come along with characteristics, fitting very well to an ontology based data management and form the preconditions for further intelligent process contributions. In this way, the process data can be used to create new information by simple inference mechanisms, exploiting the accessible knowledge. Moreover, the semantic description of the relations between data and activities can be utilized for a sophisticated process visualization as it will be shown within this paper.

We do not only consider process visualization as a possibility, but as well as a requirement. As soon as AI techniques play an active role in the execution of a process, the division of labor with human actors requires a common understanding of the process subjects. The business process visualization is the connector in this human-machine communication.

In the following we present a BPV approach based on a knowledge-based system. Therefore, section 2 introduces the elementary workflow principles. Additionally, the state of business process visualization concepts is presented. Section 3 motivates the model for a unified approach for business process modeling and execution. The capabilities are explained in detail by using an example. Section 4 introduces a new concept of a semantically oriented BPV for the unified workflow approach. Our motivation is expressed and the requirements, the architecture as well as view adaptation methods are presented. We conclude our paper by giving an outlook on our future research in Section 5.

2 Foundations

In the following, we briefly summarize relevant previous research related to business process modeling, execution, and visualization.

2.1 Business Process Modelling and Execution

Business Process Model and Notation (BPMN) is up-to-date the de facto standard for designing and describing business processes world-wide. In the center of this approach reigns a control-flow coordination of process-steps (activities). A less restrictive, but still activity-centric perspective is supported by constraint-based approaches [15], which allow flexibility in a scalable manner. Alternatively, there are several approaches with the intention to gain flexibility based on the control-flow principles [17, 13]. Despite of the consideration of data-flow in such processes, the data is just integrated in a kind of an afterthought [6, 4]. Opposed to this, knowledge intensive processes are usually barely structured and the execution is driven by user decisions and business data. Previous research has
shown [19, 2, 14] that an activity-centric perspective is not sufficient to achieve such knowledge intensive business goals.

With view to these insights, several new approaches were brought up during the last decade, putting the data into the center, not only for the design but also for the execution phase of the processes. The case handling paradigm [19] elevated the result of a process (case), reflected by its data objects; activities do not longer drive the process but serve the outcome. For more complex scenarios with the need of abstraction capabilities, object-awareness approaches refined the case handling concept [20]. With business artefacts [2], CorePro [14], and PHILharmonicFlows [11] there are even more approaches to mention which underline the importance of data-centric approaches for knowledge intensive business goals.

The existing workflow principles can be differentiated regarding the rationale for selecting activities for execution. Under a control-flow, the activities are chosen for the execution primarily by the connected ancestors, while a constraint-based model selects the activities by considering a set of restrictions. Both principles put the activity into the center of the view. This changes with the data-centric approaches, where activities are executable as soon as the necessary information is available and the expected outcome on a new information is still required for further process activities. This is expressed by the taxonomy of workflow principles, shown in Fig. 1.

![Fig. 1. Workflow principles](image)

The data-centric as well as the constraint-based concepts have in common that in both cases the relations between objects are described, while the execution order is deduced on the fly. This represents a declarative workflow definition, while the control-flow explicitly describes the execution order showing its imperative character.

The nature of both declarative principles is their inherent flexibility. Nonetheless there are major differences. With a data-centric approach, possibilities for the execution of activities are described, which support the requirements of knowledge intensive processes [7]. In contrast to this, constraints define the re-
strictions between activities, building the foundation for controlling and observing compliance rules. In a nutshell, data-centric principle defines the GO’s, while constraints are beneficial to describe the NO-GO’s. In this paper, we argue that both principles can be combined to build a unified approach, since both base on a declarative paradigm. This new approach will be introduced in section 3.

2.2 Visualization

The importance of visualization within all fields with a human-computer interaction is well established [5] and is subject to research in many segments. This paper considers mainly the field of business process visualization [3, 18, 10] (in the following denoted BPV).

The most established toolset to model business processes (BPMN) comes along with a detailed graphical notation definition. Graphs are built during the modeling phase and are usually used directly, when it comes to a process visualization later on. It represents the perspective of a process designer, which in general does not fit to the demands and needs of a process controller or an actor during the process execution. Additionally, since the modeling procedure is done to create a process template, the temporal situation of a process instance is usually just reflected by a state presentation and has no structural impact on the graph. An example of structural changes according to the process-state was introduced in Proviado [3]. With the VisModel an adaptable BPV framework was introduced and developed which offers a flexible and adaptable view on a process instance. Another flexible visualization mechanism was introduced by Jablonski and Götz [9] with a perspective-oriented process modeling approach, dividing the presentation of a process into different abstract perspective views. With the state propagation patterns [18] the challenge of different abstraction levels between the model layer EPC, BPMN, and the execution layer BPEL was addressed with the goal to transfer a process-state correctly into an abstract process presentation. ProView [10] is dealing with the special challenges of process aware information systems (PAIS) based process visualization and the need to pass changes to underlying process engines. Even if the mentioned research has introduced interesting tools and methods to go beyond the static process graph created by the process designer, all have in common that they are based on the control-flow oriented approaches like BPMN. A data perspective is available only as an add-on to the dominating activity-centric principles.

2.3 Visualization Factors

One purpose of a visualization is that the view should support the viewer in the best possible way to fulfill his/her tasks. Sophisticated business process visualization approaches [9,10] achieve this by orienting the presentation to factors like the user perspective, the process-state, as well as the personal focus.

The users in a business process are playing a certain role like: a process designer, a manager, a process controller, or a process actor. Depending on their
role, users have different user perspectives on a process to fulfill their individual tasks, which makes it to an important factor for any process adaptation.

Another factor for visualizing a process instance is the process-state being the result of the state of each process element. The importance of each element for a process visualization is mainly affected by its current influence to the process execution.

Both factors are inherent to the situation (user perspectives and process-state) and cannot be controlled by the user directly. If a visual presentation shows a process in a modified way, the user might want to have influence on this presentation by adding his personal focus to the view. This can be expressed by an interaction with the BPV and adds a third visualization factor.

These three factors claim an influence on a process visualization with different reasons and describe the points of interest for an adapted BPV.

Both, the mentioned research about business process modeling and execution and also the visualization approaches build the foundation for our work and lead us to the new concepts presented in the section 3 and 4.

3 A Unified Approach

In the following we will introduce a unified approach which offers a high flexibility during a process execution and serves the demands for knowledge intensive business processes [7]. By the integration of an ontology we take a knowledge-based system as a basis for this new approach. The artificial intelligence technique is supposed to offer some significant process contribution.

3.1 Motivation

As described in section 2.1, the data-centric and the constraint-based approaches follow the same declarative paradigm. In this work, we argue that both principles can be combined to a unified approach. We expect that this will offer a seamless scalability regarding flexibility and strictness, from an unstructured process task-list up to a narrow restrictive process model. The approach would also serve the demands of knowledge intensive processes by integrating data as a first-class citizen into the process [2]. Finally, we see the possibilities to assure the existing enterprise compliance rules by explicit restrictions based on additional activity constraints. With our work in the SEMAFLEX3 [8] project, we have already considered this by combining flexible workflow management and knowledge-based document management. Through the combination of both approaches a semantic integration based on an ontology could be achieved. With the help of document classification and information extraction methods, the process relevant data within documents is transferred to the common knowledge base. By utilizing the knowledge base, the document data allows the identification of the associated process instance and corresponding process activities.

---

3 SEMAFLEX is funded by Stiftung Rheinland-Pfalz für Innovation, grant no. 1158
Thereby, the documents can be used to recognize deviations from the current process-state and adaptations can be made to resynchronize the process.

With the unified approach, we do not only allow data to influence a process instance but to directly control its execution. Like in similar approaches [12] the demand for information within a process specifies the set of activities which can yield this information. As a result, all such activities whose preconditions are satisfied may be candidates for execution. From the data-centric perspective the preconditions are the availability of the required input information. However, through the combination with the constraint-based principle an additional layer for preconditions becomes possible. Such constraints may define direct dependencies between activities and may even guarantee a predefined execution order which can be used to fulfill compliance requirements.

Nonetheless, data has an essential influence on the process execution and can be delivered by manual activities, system activities, external sources like documents [8] and can be inferred based on the ontology. This mechanism will be explained in detail by the following example.

### 3.2 Example

Figure 2 shows a segment of an order process which delivers the information, whether a customer should be served on account or whether an upfront payment is required. The most important part in this example is the required information (Upfront payment), which is represented by a tristate value (true, false, unknown). All data elements are shown as a circle while the activities are presented in rectangles. Activities can be distinguished between manual activities like decide credit ranking manually and service activities. Edges between data and activities define the direction of the data flow. The edges between activities represent additional constraints. The transfer of an activity result can be obligated (solid lines) or optional (dashed lines).

![Fig. 2. Segment of an order process](image-url)
The process is mainly data-driven as the activities depend on the input and the output data. This means that the activities are executable if the potential output data (Upfront payment) is still required and as soon as the necessary input data (Order value, Customer) is available. Since two activities can provide the required output, an additional precedence constraint [15] is defined. This constraint is used to assure that the decision regarding the payment condition is made in a controlled and predefined order, following the enterprise compliance rules. In the following, three different cases for this process segment are discussed and present the basic mechanisms of the introduced unified approach.

Case 1: Enterprise Knowledge. In the first case, we assume that the required information Upfront payment is available right from the beginning. The information might be delivered from outside by a document. However, besides such an explicit information source there is also another option to gain that information. The customer might be known through previous order-processes and his payments were always according the payment conditions. Thus, the customer is credit-worthy, which is stored in a global knowledge store. Now, the information Upfront payment could be deduced by the integrated knowledge base, utilizing the underlying ontology which combines the process data with a global knowledge store. This way and without an explicit activity, the results, produced by the inference mechanisms, have an impact on the further execution of the process. No matter if the answer for an Upfront payment is yes or no, since the required data is already available, none of the two activities which have this information as an output needs to be executed.

Case 2: Check order value. In the case that the customer is new and thus no information about his credit-worthiness exists in the global knowledge store, the two activities are potentially executable from the data-perspective. Because of the additional constraint, only the activity check order value can be executed. As a system-activity, this can be performed without further user interaction. We assume that an upfront payment is required for any unknown customer if the Order value is above a certain threshold. In this case, the information Upfront payment is true and all other activities become obsolete again. If the Order value is below the level, the activity is successfully executed without delivering an answer for Upfront payment and the next activity can take over.

Case 3: Manual decision. This case happens if there is a new customer without knowledge about his credit-worthiness in the global knowledge store, who places an order above a certain order value. Now, all preconditions including the precedence constraint are fulfilled for a manual credit decision. A process actor, who is allowed to perform this activity, is asked for a decision, which can be true or false. However, since this is the last possibility to get an answer, this activity is obligated to deliver a result (represented by the solid line) in case of its execution.
The example has presented the data-oriented enactment of activities of the unified approach. In the execution, it follows the information-state rather than a predefined control-flow and offers a high degree of flexibility. To supervise the flexibility, constraints are used to guard predefined compliance rules.

The briefly presented unified approach combines the two declarative paradigms with a knowledge-based system. Besides the possibilities for an artificial process contribution, the ontology will be utilized for the following BPV approach.

4 A Semantically-Oriented Business Process Visualization

As presented in section 2.2, so far the research for adapting business process views focuses on control-flow oriented approaches like BPMN. Some studies have introduced adaptation techniques [3] like reduction and aggregation steps for the visualization of process instances considering process-state and user demands. However, these adaptation techniques cannot be easily transferred to the described unified approach because of its declarative and data-oriented principle.

4.1 Motivation

In general, the proposed unified approach requires a process visualization which allows to explore and identify the expected possibilities and allows the different groups of users to understand the process in its definition and its behavior during the execution. Primarily we see this demand for three user-groups, each with its own purpose:

- The process designer requires a view supportive for the process definition.
- The process controller requires a view which allows efficiently to identify delay in the process execution or potential risks.
- The process actor requires a view that serves best by the execution of single activities.

Beside the fact, that in the presented approach the relation between data and activities has a dominating role and a new visualization approach is required, the integrated knowledge base offers also new opportunities for a process view. The point of interest can be ascertained for each of the named groups and an adapted process visualization can be generated for each purpose. By utilizing the ontology, we pursue to create a view which presents the semantic relations between data and activities rather than just taking advantage of an activity arrangement by an imperative approach.

Considering the possibilities for deduced information mentioned above, humans have the need for a comprehensible presentation of the artificial process contribution. Since such new information is deduced based on the ontology, the source can be determined and the explanation can be presented by the visualization.
4.2 Visualization concept

The three introduced visualization factors (section 2.3) user perspective, process-state, and personal focus build the foundation of our visualization concept and define the individual points of interest together. These points of interest (explained in detail in the next section 4.3) and the knowledge base will be utilized to generate an adapted process view. Once the visualization is shown, three possible changes can take effect and will result in an updated process view. The user can change the process data directly or by performing an activity. The user can express his interest for a process element, which will change the personal focus. Finally, an external event (other user, system activities) can occur and the process data is changed as well.

To a large extent, this concept is similar to existing approaches. Even the user perspective and the view generation[3] are presented before in one way or another. The new aspect in this concept is the underlying data-centric paradigm as well as the knowledge base, which will be used for the view adaptation.

4.3 Points of Interest

A point of interest (POI) represents a process element which has some particular importance for the process visualization. Since the unified approach follows the data-oriented principles, a process element can either be an activity or a data-element. Any element which was recently changed or which is waiting for a change like executable activities has some elevated importance. With view to the user perspective, any activity which is assigned to the current user has also an elevated importance. If both factors (user perspective and process-state) coincide on the same point of interest, this can increase the importance even further. Additionally, through the personal focus the user can express his impression of importance. Thus, we get a list of points of interest (POIs) which filter the most important data and activity elements for the following process presentation.

The relevance of each further element depends on the individual meaning related to each POI. This meaning is reflected by the integrated knowledge base and the ontology can be utilized to calculate the relevance for each element. In a nutshell, the closer an element is related to a POI and the more meaningful the element is for the execution or understanding of the POI, the higher is its relevance. Such a relevance value can be used for each activity and data element for the following process adaptation.

4.4 View Adaptation

The view adaptation is the central transformation step of the view generator. With different techniques like reduction and aggregation, single elements up to process segments can be transformed and thus can be presented with a variable granularity. The most relevant elements can be presented in the highest level of detail while the less important elements can be presented in a more abstract view with a lower granularity. In the following, different adaptation methods
are introduced, which are partially already examined [3] under the control-flow paradigm. These methods will be transferred to the introduced unified approach and take further advantage of the integrated knowledge base.

**Reduction:** One possibility to lower the granularity of a segment is the reduction step where single elements are taken out of the view. In existing BPV approaches often system activities where reduced with the expectation that they are less important than user activities. With the introduced approach, the knowledge base is used to calculate the individual relevance in semantic relation to the POIs.

Assuming that a specific data-element (D1) has just enough relevance for an unchanged presentation, the producing activity of this information might be not important enough and thus maybe reduced from the presentation. Unless the user is not clicking on D1 and thus increasing the relevance of this element by adding his personal focus, the producing activity remains invisible in the process visualization. Alternatively, only the producing activity of the specific information is presented and all alternative potential sources, like activities, which were not executed, are reduced.

**Aggregation:** Another option to reduce the granularity is to aggregate a group of activities or data elements to a single representative element. Unlike a reduction step where the elements are completely removed, the representing symbol is still visible and accessible and the user can expand it to the fine granularity any time. In existing BPV approaches, the aggregation step is used to combine activities which follow a narrow route within the control-flow and do not split to further activities. Under this new approach, elements can be grouped and aggregated by a similar meaning. For example, data-elements with the same type of relation to a common activity can be replaced by a single element, which substitutes the individual data-elements in the visualization.

Assuming that several data-elements (D1, D2, D3) were required for an already executed activity (A1), these data-elements share the same type of relation to the activity (A1). By replacing the data-elements with a single element (D1-3) the granularity of this process segment can be reduced.

**Expansion:** The most obvious way to adapt a view is lowering the granularity of process segments. However, with view to the underlying ontology and as described in section 3.1 with the unified approach, new information can also be deduced by inference steps. Referring to the example in section 3.2, the creditworthy might be stored in a global knowledge store. This source of information is not represented in the process as an explicit activity, nonetheless it has some impact to the process execution. With the method of expansion, the process view can be expanded by further elements which are defined in the ontology and which are not an explicit part of the process definition.

Assuming that a data-element (D1) is deduced from existing data (D2) by utilizing the relations defined by the ontology, the process view can expand D1
with a further data-element D2 by representing the common relation through an additional edge.

The basic idea of the view adaptation is to gain attention for the important elements of a process according to the POIs. With reduction and aggregation, we presented two methods to lower the granularity of process segments to achieve a simplified visualization by keeping the details of the important elements. With expansion, we presented a method to add further details to the important elements to get transparency for an artificial process contribution. For these three adaptation methods as well as for the calculation of the relevance of each process element we have utilized the integrated knowledge base.

5 Conclusion

In this paper, we have shown that the data-centric as well as the constraint-based approaches follow the same declarative paradigm and can be combined to a unified approach. For this, we expect a scalable flexibility, see the demands of knowledge intensive processes fulfilled, and can use constraints to define explicit restrictions to satisfy compliance rules. Through the integration of a knowledge based system, artificial intelligence methods should be able to participate directly in a process execution. This allows a division of labor between humans and an AI system. For a common understanding and a comprehensive presentation of the unified approach a new concept for business process visualization was presented.

Referring to existing research we have considered three visualization factors (process-state, user-perspective, personal focus) to determine the points of interest in a process instance. These POIs are used as a guideline and by utilizing the integrated knowledge base, the specific relevance for each process element can be calculated. With the help of three methods (reduction, aggregation, expansion) the granularity of process segments can be adjusted according to the calculated relevance. The adaptation of the process view is taking advantage of the integrated knowledge base and a semantically-oriented process visualization is conceivable.

In our ongoing work, we will implement the described unified approach in a prototype and will define the data-driven process based on an ontology. The described architecture will be realized to transfer a process instance to a visual representation. The methods for a process adaption will be realized to prove the concept of a semantically-oriented process visualization.

References

Irreführende mentale Modelle beim Smart-TV
Misleading mental models for Smart-TVs
Gamze Ucar, Andrea Kohlhase
Information Management, University of Applied Sciences Neu-Ulm

Abstract Information Management comprises many different aspects - among others the communication of information in several modalities within the field of Human-Computer Interaction. Such interactions are experienced with varying degrees of complexity - for instance, when integrating modalities. A design goal is the reduction of interaction complexity as much as possible. Mental models help to simplify interactions and thus, are a means to tackle complexity. In this paper, we report on a study, that showcases a misleading use of mental models when dealing with smart TVs. Quite the contrary, here, the interaction complexity increases because of mixed-up mental models.


1 Einführung

Schon in den 80er Jahren gab es die Vision vom Fernseher als vernetzte Entertainment- und Steuerzentrale in den eigenen vier Wänden (siehe z.B. [LMB15]). Durch die Digitalisierung von Inhalten und Übertragungstechniken ist dies mit neuen Technologien nun möglich. Seit 2009 können Geräte der Unterhaltungselektronik mit dem Internet verbunden werden. Die Zauberwörter der Fernsehindustrie lauten „Hybrid-TV“ und „Connected-TV“, also vernetztes Fernsehen (siehe z.B. [Con15]). Moderne Fernsehbildschirme, Blu-ray-Player und auch Digital-Receiver ermöglichen die „smarte“ Nutzung, d.h. die Nutzung von zusätzlichen Services über die Verbindung zum Internet per LAN oder WLAN. Im Folgenden bezeichnen wir solche als Smart-TV's. Insbesondere ermöglichen smarte TV-Geräte die Vernetzung mit anderen Unterhaltungselektronikgeräten. So sind am Bildschirm digitale Fotoapparate, Kameras, DVD- sowie Blu-ray-Geräte, Spieleskonsolen, MP3-Player und sogar Smartphones nutzbar. Es gibt inzwischen Lösungen für die Steuerung des Fernsehers über Smartphones und Tablets durch Apps (siehe Abbildung 1).

Interessanterweise ist die Internetnutzung bisher an Computer bzw. Smartphones gebunden, das Fernsehen aber an TVs. Für die verschiedenen Medien werden typischerweise bei der Benutzung unterschiedliche mentale Modelle verwendet. Ein mentales Modell ist ein Modell, also eine Reduktion auf wesentliche Eigenschaften, der Benutzung eines Artikats für einen Benutzer (siehe [Joh80]). Insbesondere sind mentale Modelle Hilfen bei der Benutzung, die schon vor der eigentlichen Benutzung die Art und Weise wie ein Artikat benutzt werden sollte, prognostizieren und so die Erstellung von Aktionsplänen ermöglichen. Zum Beispiel müssen wir nicht wirklich (alle) verstehen, wie ein elektrischer Lichtschalter funktioniert. Es reicht beispielsweise, wenn wir uns Strom als fließendes Material vorstellen und den Lichtschalter als eine Art Sperre, um ihn (oder dann auch z.B. Strom-Steckdosen) leicht benutzen zu können.


Uns interessiert hier besonders, ob die mentalen Modelle für die Bedienung von klassischen TVs mit denen für Computer beim Smart-TV kollidieren oder sich ergänzen.

1 Im Folgenden nennen wir das mentale Modell, welches die Metapher eines internetfähigen Computers (Laptops, Tablets, Smartphones, ...) im weiten Sinn benutzt “Computer”.
<table>
<thead>
<tr>
<th>Methode</th>
<th>Beschreibung</th>
<th>Vorteile</th>
<th>Nachteile</th>
</tr>
</thead>
</table>
2 Die explorative Studie


Um die Interaktionskomplexität eines Samsung Smart-TVs genauer verstehen zu können, wurde im Testplan (siehe Abb. 3) auf verschiedene Fragen eingegangen. Dabei wurden die Aufgaben in Form von Szenarien gestellt, damit sich die Testpersonen in die Alltags situation besser hineinversetzen können und somit ein wenig die Laboratmosphäre aufgelöst wird.

3 Ergebnisse

Bei der Auswertung des Tests wurden jeweils gleiche oder ähnliche Antworten mit Hilfe der Card-Sorting-Methode zu einem Oberbegriff zusammengefasst. Wir gehen auf die Ergebnisse jeder Aufgabe im Testplan in Abb. 3 im Folgenden gesondert ein.

3.1 Aufgabe 1

Die für Aufgabe 1 gesammelten Eye-Tracking-Daten wurden als Heatmap (stellvertretend von 5 Probanden) in Abb. 4 visualisiert. Es ist bekannt, dass ein prädestinierter Anfangsblickpunkt für
Abbildung 3: Testplan der Studie

eine Suche auf einer Website mit unbekanntem Inhalt die Mitte derselben ist. Da beim Smart-TV ja bewegte Bilder gezeigt werden, ist die Orientierung schwieriger, so dass eine allgemeine Bevor-
zugung der linken Seite bei den Probanden durch ihr normales von-links-nach-rechts-Lesen nicht verwundert.


3.2 Aufgabe 2

In der zweiten Aufgabe mussten die Testpersonen unter den Samsung-Apps (Meine TV-Apps) Facebook finden und herunterladen. Die Samsung-Apps sind unter der Rubrik „Empfohlen“ (siehe Abb. 5 und 6) zu finden. Das heißt, die Samsung-Apps sind etwas versteckter und nicht auf den ersten Blick im Smart-Hub auffindbar. Das Finden der Samsung-Apps war für alle sechs Smart-TV-Experten kein Problem. Sie wurden ebenfalls innerhalb von wenigen Sekunden gefunden.


3.3 Aufgabe 3


3.4 Aufgabe 4

Wenn die Testpersonen nicht eigenständig herausfanden, wie eine Sendung aufgenommen werden kann, wurden ihnen zwei Hilfestellungen gegeben. Der erste Tipp lautete: „Es gibt eine Gebrauchsanweisung, in der du nachlesen kannst, wie die Aufnahme funktioniert“ und der zweite Tipp lautete: „Es gibt für die Gebrauchsanweisung eine Taste auf der Infrarot-Fernbedienung“. 
Fünf der Novizen und drei der Experten konnten trotz der zwei Hilfestellungen die Aufgabe nicht lösen. Vier von ihnen gaben an, dass sie in einer komplexen Situation auf Youtube oder Google nach einer Lösung suchen würden. Hier werden also Webdienste zur Problemlösung bei der Nutzung eines Smart-TVs herangezogen, aber interessanterweise nicht auf dem internetfähigen Medium selbst.

3.5 Aufgabe 5

3.6 Aufgabe 6
Zu der Frage „Was geht dir bei der Betrachtung der Fernbedienungen durch den Kopf?“, haben sich die Aussagen in Abb. 8ergeben (die Zahl hinter den Aussagen gibt die Anzahl der Personen an, die mit gleicher oder ähnlicher Aussage geantwortet haben): Die Größe der Infrarot-Fernbedienung hängt mit der Anzahl der Tasten zusammen. Dadurch wird das Erreichen aller Tasten schwieriger. Die Benutzer müssen mehr auf- und ab Bewegungen machen um an die gewünschte Taste heranzukommen. Daher wird die Pointer-Fernbedienung, mit weniger Tasten, als kompakter und handlicher empfunden. Die Optik der Pointer-Fernbedienung

<table>
<thead>
<tr>
<th>Infrarot-FB</th>
<th>Pointer-FB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zu groß</td>
<td>Klein</td>
</tr>
<tr>
<td>Zu viele Tasten</td>
<td>Weniger Tasten</td>
</tr>
<tr>
<td>Zu viel Text</td>
<td>Fradischer</td>
</tr>
<tr>
<td>Suche nach gesuchter Taste dauert lange</td>
<td>Moderner</td>
</tr>
<tr>
<td>Übersichtlich</td>
<td>Kompakt</td>
</tr>
<tr>
<td>Altmöden</td>
<td>Alle nötigen Tasten vorhanden</td>
</tr>
<tr>
<td>Vertraute FB</td>
<td>Handlicher</td>
</tr>
<tr>
<td>Alle Tasten schwieriger zu erreichen</td>
<td>Einfacher</td>
</tr>
<tr>
<td>Zum Surfen unkomfortabel</td>
<td>Mit Pointer-Funktion einfacher alles zu erreichen</td>
</tr>
</tbody>
</table>

Abbildung 8: Aufgabe 6 – Kategorisierte Antworten auf “Was geht dir bei der Betrachtung der Fernbedienungen durch den Kopf?”
trifft auch den Geschmack der Probanden. Da mit dem Pointer das Steuerkreuz zur Auswahl von Menüpunkten oder Buchstaben auf der On-Screen Tastatur nicht genutzt werden muss, wird sie als einfacher und praktischer bewertet.


3.7 Fragebogen

Der erste Fragenkomplex beschäftigt sich mit den von den Testpersonen benutzten Analogien. Der zweite Fragekomplex hingegen, beschäftigt sich mit der subjektiven Wahrnehmung der Komplexität des Zusammenspiels verschiedener Komponenten. Es ist auffällig beim ersten Fragekomplex (siehe Abb. 9), dass die Mehrheit der Probanden die Pointer-Fernbedienung mit einer Computer-Mouse, aber nicht mit einer Computer-Tastatur vergleicht (dies scheint aber auch nicht notwendig, denn eine ausklappbare Tastatur wünschen sich die Probanden nicht). Trotzdem empfindet die Hälfte der Probanden das Smart-TV als eine Mischung aus Laptop und selbstgesteuertem Kino. Die Probanden unterscheiden jedoch recht deutlich zwischen einem Tablet und Smart-TV.

Wir gehen davon aus, dass durch die Analogiebildung mit einem geeigneten mentalen Modell die Interaktionskomplexität reduziert wird und dann also die Aufgaben leichter gelöst werden sollten. In unserem Fragekomplex 1 haben wir nur nach bestimmten Analogien gefragt. Wir setzen dabei voraus, dass, wenn die Antwort „trifft zu“ gewählt wurde, der Proband auch das entsprechende mentale Modell bei der Benutzung herangezogen hat, wogegen, wenn die Antwort „trifft nicht zu“ gegeben wurde, der Proband ein anderes mentales Modell verwendet. Da wir mit dieser Studie sowieso nur Tendenzen vermuten können, haben wir in der Diskussion die „trifft zu“- und „trifft eher zu“-Zahlen zu „benutzt traditionelles Modell“ zusammengefasst und den Rest zu „benutzt anderes Modell“. Um die Zahlen besser miteinander vergleichen zu können, haben wir weiterhin die Zahlen aus Abb. 9 in Abb. 10 vereinfacht und als Prozentzahlen angegeben. Aus Abbildung 10 können wir ablesen, dass nur 38,9% aller Probanden zumindest teilweise ein traditionelles mentales Modell bei der Benutzung herangezogen haben. Da hier die Abweichungen besonders groß sind, differenzieren wir nochmal: Die
Fernbedienung wurde von 75% mit einer Maus verglichen, nur von 16,7% als Tastatur und nur 8,3% verglichen ein Smart-TV mit einem Tablet-PC. Hieraus könnte nun eine erste These abgeleitet werden, nämlich die, dass eine Pointer-FB versucht wird wie eine Maus zu bedienen. Insbesondere die Zeige-Funktionalität einer Pointer-FB kann mit der der Maus verglichen werden. Diese Analogie wird z.T. auch visuell durch die Form der Pointer-FB (siehe Abb. 3) unterstützt. Allerdings hat eine Maus nur zwei bis drei Tasten, eine Fernbedienung aber typischerweise (sehr viel) mehr. Daher liegt auch die Metapher der Tastatur als mentales Modell für die Verwendung einer Fernbedienung nahe. Allerdings sehen das von unseren Probanden ja nur 16,7% so. Schon hieraus erkennen wir, dass das mentale Modell für eine Fernbedienung nicht so einfach von den bisherigen Modellen abgeleitet werden kann.

Nun schauen wir uns unsere Daten bzgl. des ersten Fragenkomplexes mit Hilfe der Abbildungen 11 und 12 genauer an. Interessant wäre es ja, wenn wir in unseren Daten Anzeichen dafür fänden, dass die Lösbarkeit der Aufgabe mit diesen mentalen Modellen zusammenhängt. Wenn wir uns Abbildung 11 im De-
unserer Daten in Abbildung 12 zeigt, dass 33,3% der Novizen und 44,4% der Experten ein traditionelles mentales Modell benutzt haben. Wie auch schon bei den Gesamtzahlen in Abb. 10 schauen wir uns die Variation an: Die Experten verwenden bei der Bedienung des Smart-TVs mit der Fernbedienung zu 100% die Analogie mit der Maus und zwar ganz explizit (d.h. keine tendenzielle “trifft eher zu”-Aussage). Dagegen empfinden die Experten ganz explizit den Vergleich eines Smart-TVs mit einem Tablet zu 100% als nicht richtig. Beachtet werden muss hierbei, dass nur ein Novize und nur die Hälfte der Experten die Aufgaben gelöst haben. Wie in den Unterkapiteln 3.1 und 3.2 beschrieben wurde, ist aber der Label-Content an der Metapher Computer bzw. Smartphone-Apps ausgerichtet. Hier gibt es also auch deutlich irreführende suggerierte mentale Modelle.

Aus den Ergebnissen wird deutlich, dass die Probanden sich sehr uneinig sind und das Smart-TV noch nicht als Medium zum Online Surfen ansehen und nicht in eine bestimmte Kategorie einordnen können. Offensichtlich ist das Smart-TV, trotz des PC ähnlichen Bildschirms, prinziell ein anderes Medium.

 Abordnung 12: Fragenkomplex 2

Beim zweiten Fragenkomplex in Abb. 13 stellt sich heraus, dass unabhängig davon, ob die Testperson ein Smart-TV-Novize oder Smart-TV-Experte ist, die Bedienung mit dem Pointer als sehr angenehm bewertet wird. Dabei ist zu beachten, dass aber nicht alle Aufgaben ohne Hilfe gelöst werden konnten. Zwei Testpersonen (Smart-TV-Novizen), empfanden die Interaktion zwischen der Fernbedienung und dem Samsung TV als komplex. Diese zwei Testpersonen empfanden sie auch als verwirrend. Drei Testpersonen, (Smart-TV-Experten), die die Interaktion zwischen der Fernbedienung und dem Samsung TV als nicht komplex empfanden, waren der Meinung, dass die Interaktion nicht verwirrend, einfach, selbsterklärend, natürlich und integriert ist. Es kann also zusammenfassend ge-
sagt werden, dass Smart-TV-Novizen die Interaktionskomplexität des Smart-
TVs in unserer Studie anders wahrnehmen als Experten. Der Erfahrungswert
spielt also eine wesentliche Rolle bei der Beurteilung.

4 Zusammenfassung

In diesem Papier haben wir uns mit der Interaktionskomplexität von Smart-
TVs befasst und insbesondere mit den mentalen Modellen, die diese reduzieren
können. Dabei haben wir festgestellt, dass die Lernkurve bei Smart-TVs steil
tzu sein scheint, da nicht viele Probanden die Aufgaben überhaupt vollständig
lösen konnten. Da unsere Aufgaben nicht besonders anspruchsvoll waren, können
wir hieraus auf jeden Fall schon ein Usability-Problem ableiten. Bei genauem
Hinschauen haben wir festgestellt, dass die von den Probanden verwendeten men-
talen Modelle zum einen noch nicht eindeutig sind, zum anderen aber auch
nicht adäquat. Insbesondere werden auch mentale Modelle wie das “Computer”-
Modell zur Benutzung herangezogen, die in die Irre führen, weil die Smart-TV-
Produzenten nicht so sehr den Benutzer, sondern die Marketingstrategien für
ihre eigenen Produkte im Blick haben.

Unsere Studie ist eine kleine, explorative Studie, die noch keine allgemeingülti-
gen Aussagen zeigen kann. Aber der Zusammenhang zwischen den mentalen Mo-
dellen und der Interaktionskomplexität bei der Benutzung von User Interfaces
kann als Ausgangspunkt für weitere Studien genommen werden. Insbesondere
wäre die Frage interessant, inwiefern User von der Verwendung alter menta-
taler Modelle abzubringen sind bzw. wie diese sinnvoll in neue integriert werden
können.

References

[Con15] Deloitte Consulting. “Video interaktiv”. In: Deloitte Media Consumer Sur-
general role for analogical encoding”. In: Journal of Educational Psychology
95.2 (2003), pp. 393–408.
[Joh80] Philip N. Johnson-Laird. “Mental Models in Cognitive Science”. In: Cogni-
URL: http://dx.doi.org/10.1207/s15516709cog0401_4
der Consumer Electronics - 2015: Marktentwicklung, Schlüsseltrends, Me-
URL: https://www.bitkom.org/noindex/Publikationen/2016/Leitfaden/
CE-Studie-Update/160226-CE-Studie-2015-online.pdf
Neu-Ulm University of Applied Sciences, 2017.
Workflow Flexibility by Deviation by means of Constraint Satisfaction Problem Solving

Lisa Grumbach, Ralph Bergmann
University of Trier, Department of Business Information Systems II, 54286 Trier, Germany

Abstract. This paper introduces a novel approach for flexible workflow management by applying constraint satisfaction problem solving. This enables us to support workflow deviations at runtime, react to upcoming events or unpredictable circumstances, but still support the user through worklist suggestions. The developed workflow engine is completely based on declarative workflow representations, whereas procedural languages are used for workflow modeling.

1 Introduction

In small and medium-sized enterprises (SMEs) there is a strong demand for support concerning management of documents, business data, and processes as well as a need for supervision and control of all running and completed transactions [14]. Especially for employees who are unaware of common processes, a Process-Aware Information System (PAIS, [1]) would be of advantage, as they may profit from guidance concerning ideal workflow execution and task suggestion. Additionally, compliance concerning standards and guidelines would be facilitated, as benefit for the enterprises. However, the use of PAISs has not yet been broadly established in SMEs. A reason for this is that current PAISs control process execution by traditional workflow engines in which workflows are prescribed without providing any flexibility to deviate, if necessary [5, 15]. This is a particular problem in SMEs as their processes are only slightly standardized and weakly structured and may vary significantly from case to case [9, 17].

Artificial Intelligence (AI) is a key technology for various support strategies in Business Process Management (BPM), as it allows for automated decision making and thus, facilitates the users work. Allowing flexibility requires such intelligent technologies, as the user should only be guided executing a workflow and not be burdened with taking difficult decisions that could be automated. Declarative workflows are a means of implicitly offering flexibility but therefore require technologies from the field of AI for workflow control. DECLARE [2] is a tool suite for declarative workflow modeling and enactment. Declarative models consist of constraints which define undesired behaviour. Constraints are then transformed to finite-state automata which allow for reasoning about workflow states. A drawback of this approach is that this transformation process is inefficient for more than about 50 constraints [10] and thus runtime support for
changing circumstances is not provided. With an approach based on constraint satisfaction problem (CSP) solving we aim at achieving model changes at run-time efficiently, as constraints can simply be added or retracted without the need of a transformation. Furthermore with a combination of imperative and declarative paradigms the presented approach leads to an increased flexibility.

In this paper we present an approach for flexible workflow execution utilizing CSP solving to handle occurring deviations and to control worklist suggestions. First, the foundations concerning flexible workflow management are sketched, followed by the introduction of our new concept for combining imperative and declarative paradigms for flexible workflow execution. This approach is further described by algorithms which are based on CSP solving. The paper ends with a brief outlook on future work.

2 Foundations and Related Work

A workflow is “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules” [21]. A Workflow Management System (WfMS) supports the execution of workflows by a workflow engine that interprets the process definitions and interacts with a worklist handler, which is in charge of assigning work items to users.

Workflow Flexibility

Traditional WfMS are rigid and do not allow any deviations from modeled workflows. Users feel restricted and such systems are rapidly considered as a burden. Thus, users bypass the systems, which is counterproductive for attaining the expected benefits [5]. Consequently, PAISs that allow a workflow to flexibly deviate are essential for efficiency in SMEs. Schonenberg et al. [16] distinguish between four kinds of workflow flexibility: Flexibility by Design, Change, Underspecification, and Deviation. The first three types require either complete knowledge about all possible workflow execution paths at design-time or demand a remodeling of the workflow at run-time. Hence, a flexible reaction to sudden changing circumstances during run-time is prevented, or actions are required to manually change the process instance, which is impossible for inexperienced users. “Flexibility by Deviation is the ability for a process instance to deviate at run-time from the execution path prescribed by the original process without altering its process model.” [16]. Although this approach eliminates the previously mentioned disadvantages, little research exists on how to implement this approach. Only the system FLOWer [3] implements this idea to a limited extent by allowing the user to skip, undo, or redo a task or to insert a new task, but still the user has to intervene manually to obtain flexibility.

Workflow Modeling Paradigms

Workflow modeling paradigms range from imperative (procedural) to declarative [6]. Imperatively modeled workflows explicitly specify all possible allowed execution paths, for example using a flow-based modeling language such as BPEL ([4]). Here, the control flow of tasks
as well as the related flow of data items is modeled, which results in a high complexity and a huge modeling effort. Declarative workflows, however, define forbidden behavior and states of the workflow. Imperative workflows only describe a subset of valid procedures, while declarative constructs describe specific undesired states, leading to the acceptance of every other state [11] and thus, implicitly providing flexibility concerning workflow execution.

Current declarative workflow approaches such as DECLARE [2], formally base on Linear Temporal Logic formulae representing constraints, which are further transformed into finite-state automata, for constraint validation, as workflow engine and for worklist handling. Though there is a differentiation between mandatory and optional constraints, and optional ones may be violated, a possibility to retract constraints is not specified and therefore no unforeseen situations can be handled flexibly. The concept of DCR graphs [8] is described as offering more flexibility, but also has no possibility to restore consistency after deviations. In later work Maggi et al. [10] developed an approach, Mobucon, based on colored automata, with the ability to detect deviations and in addition to support continuously through various strategies. A drawback of this approach is that strategies need to be determined beforehand, and cannot be changed during runtime, as the construction of a new automaton would take too long [10]. Algorithms developed by Westergaard [20] also solve this issue with efficient runtime modifications, e.g. models with up to 50 constraints are handled in seconds.

Our approach also aims at achieving efficient automated runtime modifications and thus requires the ability to react adequately to deviations, even to undesired situations. A main difference between related work and our approach is that our workflow control bases on the interpretation of incoming documents and their semantic information. The identification of semantic information has a significant impact on workflow control and can be easily defined as logical constraints. Furthermore constraints can be added or retracted ad-hoc, without the need of a time-consuming recompilation of the model. Therefore we regard the identification of executable tasks as constraint satisfaction problem.

3 Concept of the Workflow Engine

With the presented concept we aim to increase the acceptance of users, as the presented concept does not prescribe, but still guides, if needed. Additionally, transactions are logged for control and monitoring purposes. The implementation of the approach of Flexibility by Deviation as presented in this paper is embedded in the SEMAFLEX¹-architecture [7], which semantically integrates flexible workflow management with knowledge-based document management and will be developed further in the SE MANAS project. An important characteristic is that the information about task enactment can either result from a user interaction, e.g. a manual selection of a task being performed, or due to upcoming documents, which are analyzed automatically and mapped to a certain task, whose

¹ SEMAFLEX is funded by Stiftung Rheinland-Pfalz für Innovation, grant no. 1158
enactment is derived subsequently. These logged task enactments construct the actually conducted workflow as a sequence of activities that have been performed. While the workflow engine proposes tasks which should be done next, the user is not forced to follow these suggestions. In principle, the user is able to do what s/he wants and in which order s/he wants. S/he can either follow the tasks in the worklist, suggesting the standard course of action, or do something else and upload documents created as a result of what s/he did. Through both, explicitly completing a task and uploading documents, the actual workflow is identified and recorded. Progress in turn affects the worklist handling, including detected deviations.

3.1 Combination of Imperative and Declarative Paradigms

For a suitable representation of the workflows concerning this concept, we explicitly differentiate between modeled workflow, de jure workflow, and executed workflow, de facto workflow [1]. The de facto workflow is an actually enacted instance derived from a de jure workflow. Thus, it stores the actually conducted transactions and thus might deviate from the de jure workflow.

In our approach the de jure workflow is modeled procedurally, as it is more intuitive and comprehensible than declaratively modeled workflows [12]. The workflow engine, however, is completely based on a declarative representation, as this paradigm implicitly offers flexibility concerning execution. To reach a maximum of flexibility, we transform the de jure workflow into declarative constraints, which are used to control the suggested execution order of tasks, but which are not regarded as mandatory and consequently might be violated. Hence, the de jure workflow is only considered as guidance, but deviations are tolerated. Nevertheless, some deviations are critical and should never occur, considering e.g. compliance or safety aspects. For this reason, additional mandatory constraints can be modeled manually, to explicitly specify invalid workflow states. Those are possibly connected with a severity specification, a warning message or even a proposed corrective measure, in case the constraint is violated. Such mandatory constraints can refer to the execution order of the tasks within a de jure workflow or they could be global constraints specifying order constraints across classes of workflows. Of course those mandatory constraints might actually be violated by the user, as the workflow engine never prescribes an activity and thus is not able to actively prevent violations. Nevertheless, the violation of constraints (including the mandatory ones) can be detected. Depending on the kind of constraint violation the workflow engine shall be able to react adequately. If a non-mandatory constraint is violated, the deviation is not considered as critical, but the workflow engine must reason about the next task to propose. If a mandatory constraint is violated, a warning is issued or a corrective measure is performed according to what is specified for the constraint.
3.2 Declarative Workflow Representation

For the declarative workflow representation, we utilize five different constraint types of the DECLARE language [2], which counteract possible deviations:

- **Precedence**($t_a, t_b$): Task $t_b$ can only be executed after $t_a$.
- **Response**($t_a, t_b$): Task $t_a$ requires the enactment of $t_b$.
- **Existence**($t_a$): Task $t_a$ is mandatory.
- **Not Co-Existence**($t_a, t_b$): Task $t_a$ and $t_b$ exclude each other.
- **Absence**($t_a, x$): Task $t_a$ can only be executed $x$ times.

Precedence and Response prevent undesirable skipping, concerning previous or subsequent tasks. Existence contradicts the undoing of a task. Redoing and creating additional instances of a task are intercepted by the constraint Absence. Not Co-Existence avoid invoking undesired task enactments.

Preventing the user from constraint violations in our case can only be achieved, if the user manually chooses tasks from the worklist, as only such tasks are proposed that lead to a valid workflow state. As non-mandatory constraints might be violated, the worklist could consider tasks that contradict these constraints but with lower priority. Mandatory constraints should never be disregarded. Worklist handling is easy for procedurally modeled workflows, if no deviations are possible. However, if a deviation occurs, one would not be able to suggest an appropriate further proceeding. Constraints suit this situation perfectly, as even if one is violated, it might be retracted, and still valid suggestions can be computed with the help of remaining constraints. How valid task suggestions are identified, will be explained in the following section.

A prerequisite for the presented enactment approach is that each construct of the de jure workflow, modeled imperatively, is automatically transformed into corresponding declarative expressions.

3.3 Transformation into Declarative Constructs

We consider the essential structures of imperative modeling languages on the basis of Weske [19]. The transformation rules, as described in Tab. 1, are defined analogously to [18], who in contrast uses Relation Algebra as formal specification language. The resulting constraints are used for validating upcoming enactment states of the workflow and for proposing tasks enabled for execution. The following section describes the algorithm that computes possible task suggestions on the basis of these declarative constraints.

4 Worklist Handling by means of Constraint Satisfaction Problem Solving

According to Russell and Norvig [13] a constraint satisfaction problem (CSP) is defined by a set of decision variables $X = \{X_1, X_2, ..., X_n\}$ and a set of constraints $C = \{C_1, C_2, ..., C_m\}$. Each decision variable has a domain $D_i$, which
Table 1. Mapping imperative workflow patterns to declarative constructs

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
<th>Corresponding Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td><img src="image" alt="Sequence Diagram" /></td>
<td>Precedence(A,B)</td>
</tr>
<tr>
<td>Xor-Split</td>
<td><img src="image" alt="Xor-Split Diagram" /></td>
<td>Precedence(A,B) and Precedence(A,C)</td>
</tr>
<tr>
<td>Xor-Join</td>
<td><img src="image" alt="Xor-Join Diagram" /></td>
<td>Precedence(B,D) or Precedence(C,D)</td>
</tr>
<tr>
<td>Xor-Sequences</td>
<td><img src="image" alt="Xor-Sequences Diagram" /></td>
<td>Not Co-Existence(A,C) and Not Co-Existence(B,C) and Not Co-Existence(A,D) and Not Co-Existence(B,D)</td>
</tr>
<tr>
<td>And-Split</td>
<td><img src="image" alt="And-Split Diagram" /></td>
<td>Precedence(A,B) and Precedence(A,C)</td>
</tr>
<tr>
<td>And-Join</td>
<td><img src="image" alt="And-Join Diagram" /></td>
<td>Precedence(B,D) and Precedence(C,D)</td>
</tr>
<tr>
<td>Cycle</td>
<td><img src="image" alt="Cycle Diagram" /></td>
<td>cf. Xor-Join and Xor-Split w.r.t. single instances of tasks</td>
</tr>
</tbody>
</table>

is a nonempty set of possible values for $X_i$. A constraint $C_j$ is a relation over a subset of the variables $\{X_k, ..., X_l\}$, specifying the set of combination of allowed values. An assignment of values to some or all of the variables is called state of the problem, which is denoted as consistent, if it does not violate any constraint. If values are assigned to every variable, the assignment is named complete. A consistent and simultaneously complete assignment is called solution. In the following we formulate the problem of selecting the next tasks for execution in case of deviation from the de jure workflow as a constraint satisfaction problem.

4.1 Application of CSP Solving

The CSP solving algorithm is applied during workflow execution at the start of each workflow and after each task enactment. The initial state for the algorithm is a partially completed workflow, the de facto workflow, and its corresponding ideal course of events, the de jure workflow. The desired output is the set of tasks, called worklist, that can be enacted next without violating constraints.

To utilize CSP solving for worklist handling, we regard the workflow tasks as decision variables $X = T$. For each task $j_i \in J = \{j_0, ..., j_{n-1}\}$ of the de jure workflow, a variable $t_{j_i}$ is created and added to $T$. Subsequently, $T$ is supplemented with one single variable $t_{\text{end}}$, to be able to determine whether the workflow has completed. The elements of this set may vary while the workflow...
progresses (see Sect. 4.3). The assignment of values to tasks represents a sequential order of all tasks, including not only already executed tasks, but also possible future executions of tasks. Thus, a valid order, determined by ascending integer values, of tasks is calculated. As the only thing of interest is, which task may be executed next at a specific point in time, it does not matter if any other tasks might be or have been executed in parallel. Consequently, the domain for each decision variable is a set of integer values \( D_i = \{0, 1, \ldots, n\} \), with \( n \) as the number of tasks extracted from the de jure workflow including the additional variable \( t_{\text{end}} \).

As the assignment represents a sequential execution order of all tasks, the first given constraint (see (1)) states that each assigned value of a decision variable is different from all others. Thus, only a bijective mapping of domain values to decision variables is a solution to the CSP.

\[
C = \{\text{alldifferent}(T)\},
\]

\[
t_i = c_i, \ldots
\]

\[
t_a < t_b,
\]

\[
(l_{\text{end}} < t_a) \lor (t_a < t_b) \land (t_b < l_{\text{end}}),
\]

\[
t_a < l_{\text{end}},
\]

\[
(l_{\text{end}} < t_b) \lor (l_{\text{end}} < t_a),
\]

\[
\sum_{i=0}^{n} f(\text{ref}(t_i), t_a) < x,
\]

\[
(s_i = a_1) \Rightarrow (l_{\text{end}} < t_2),
\]

\[
\land (s_i = a_2) \Rightarrow (t_{\text{end}} < t_1)
\]

with \( \text{ref}(t_i) \) returning the id of the referenced object of the de jure workflow and \( f(\text{ref}(t_i), t_a) = \begin{cases} 1 & \text{if } \text{ref}(t_i) = t_a \\ 0 & \text{otherwise} \end{cases} \)

Second, as we apply the CSP at a specific point in time during execution of the workflow, some tasks are already enacted and therefore the respective variables have a fixed assignment \( c_i \), which is a constant value specifying the sequential execution position in the de facto workflow (see (2)). All additional constraints either result from the transformation of imperatively modeled workflow to declarative language constructs or originate from manually modeled mandatory constraints. Each type of constraint has a corresponding formal definition to be used in the CSP solving algorithm (see (3) to (7)). As some constraint violation explicitly depends upon workflow completion, \( t_{\text{end}} \) is used to determine the termination of a workflow. This is necessary to assert a mandatory enactment of a task, a required execution of a task after a certain one, or even to assure that some tasks have not been conducted. Considering the solution of the CSP every task with a higher integer assigned than \( t_{\text{end}} \) is regarded as not enacted.

Due to the construction of the CSP, we are also able to influence the control of the workflow on the basis of semantic information. Control-flow nodes are
additionally considered as constraints to further automate the control process. On this basis tasks are excluded from enactment proposals. For each xor or loop construct (cf. Fig. 1) two constraints are included (see (8, 9)). With $s_i$ representing a decision variable, which either takes $a_1$ or $a_2$ as value, we are able to derive which path in the workflow should be followed. For workflow control the execution of the oppositional path is prevented, e.g. if the information of $s_i$ is known to be $a_1$, task $t_2$ should not be enacted ($t_{end} < t_2$).

![Fig. 1. Xor and loop construct](image)

Furthermore, considering the importance of data nodes for the presented approach, additional constraints are generated on the basis of data dependencies. For example, if data node $d_1$ is output of task $t_1$ and input of $t_2$, a constraint $Precedence(t_1, t_2)$ is inserted in the set of constraints, as it is necessary to enact task $t_1$ to subsequently process $d_1$ with $t_2$.

Table 2 shows an example transformation from procedurally modeled workflow to constraints to logical formulae, which are then used by the algorithm.

### 4.2 Worklist Handling Algorithm

To identify all tasks that might be enacted next, a solution to the presented CSP, on the basis of the previously explained generated constraints, is searched for (cf. Algo.1). The algorithm takes the sets $T$ and $C$ as input for CSP Solving, whereas the domains $D_i$ are derived from the size of $T$. The set $F$, also used as input, represents the de facto workflow and contains the variables from $T$, whose referenced tasks have been enacted. As output the variable result is introduced, which represents the set of tasks that might be enacted next. The value of the expected tasks, denoted here as current, is determined by the size of $F$, as this is the position which will be occupied by a task executed next. If every solution

<table>
<thead>
<tr>
<th>Example workflow</th>
<th>Declarative constraints</th>
<th>Logical formulae for the CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="image" /></td>
<td>$Precedence(A,B)$, $A &lt; B \land A &lt; C$\land</td>
<td>$(B &lt; D \lor C &lt; D)\land$</td>
</tr>
<tr>
<td></td>
<td>$Precedence(A,C)$, $B &lt; D \lor C &lt; D)$\land</td>
<td>$Precedence(B,D)$ or $(End &lt; B \lor End &lt; C)$\land</td>
</tr>
<tr>
<td></td>
<td>$Precedence(C,D)$, $(XY = yes) \Rightarrow (End &lt; C)$\land</td>
<td>$Not\ Co-Existence(B,C)(XY = no) \Rightarrow (End &lt; B)$</td>
</tr>
</tbody>
</table>

219
Input: T, C, F
Output: result: Tasks that might be executed next

1 \( \text{result} = \emptyset; \)
2 \( \text{foreach } D_i \in D \text{ do } D_i = \{0, \ldots, |T| - 1\}; \)
3 \( \text{current} = |F|; \)
4 \( \text{foreach } x \in T \setminus F \text{ do } \)
5 \( \quad \text{if solveCSP}(T, D, C \cup \{x = \text{current}\}) \neq \emptyset \text{ then } \)
6 \( \quad \text{result} = \text{result} \cup \{x\} \)
7 \( \text{end} \)
8 \( \text{end} \)
9 \( \text{return result}; \)

Algorithm 1: Determination of tasks which might be executed next

to the CSP would be computed, this would result in redundant and unnecessary computations. To accelerate proceedings, we apply the CSP solving algorithm only once for each decision variable that has not been assigned a value yet, thus, has not been executed. If the CSP solving algorithm finds a solution, this task might be executed next and therefore is appended to result. If no solution is found, the user must not execute the task next and thus, it is not proposed.

4.3 Deviation Detection

If a task enactment occurs, variable assignments are updated and constraints are validated to analyze the state of the workflow and possibly restore consistency. Algorithm 2 illustrates this procedure. As input, the enacted task \( j_n \) of the de jure workflow is received. First, the length of the de facto workflow needs to be determined, which identifies the assigned value to the variable of the currently enacted task within the CSP. The corresponding variable \( t_{j_n} \) is included in the de facto workflow \( F \), and constraints are extended with the value assignment of \( \text{current} \) to \( t_{j_n} \). As this task enactment might result in a constraint violation, and consequently in an inconsistent workflow state, all constraints need to be validated. If no solution is found, the violated constraint needs to be identified and retracted from \( C \) to restore consistency. In case a mandatory constraint is violated, the respective warnings and corrective actions are triggered. After each task enactment the constraint set is simplified in order to prevent unnecessary computations. Based on the value assignments due to the de facto workflow, which will never change for a workflow instance, some constraints will always resolve to true, while other parts always resolve to false, even without constraint violations, e.g. disjunctive associated propositions. Assuming that the constraint set is available in conjunctive normal form \( C = c_1 \land \ldots \land c_n \), clauses \( c_i \) are linked conjunctively and each clause represents a disjunction of literals \( c_i = l_1 \lor \ldots \lor l_n \). The literals mostly result from the transformation of declarative workflow constructs to logical representations and thus, relate two tasks with the ordering "<". Other literals may be equations, such as \( t_1 = 0 \), depicting the de facto workflow, or \( \text{alldifferent}(T) \) due to the construction of the CSP. The
Algorithm 2: Deviation Detection

literals of interest for CSP simplification are the first ones. If a task enactment of task \( t_i \) occurs, all clauses with \( t_i \) on the left side (e.g. \( t_i < t_j \)) in one of the literals are withdrawn, as this clause will resolve to true in any case. Furthermore single literals, which contain \( t_i \) on the right side, e.g. \( t_j < t_i \), can be retracted. Those will never be fulfilled, but the remaining literals in the clause have to be.

One impact of this simplification strategy after each task enactment is that violated constraints can be easily determined. A clause consisting of a single literal, e.g. \( t_i < t_j \), where the currently enacted task \( t_j \) is on the right side, is violated, as \( t_j \) has not yet been enacted, as otherwise the clause would have been withdrawn from the constraint set previously.

4.4 Extension of the CSP for Loop Patterns

Until now the approach is limited to a singular execution of tasks and not incorporating loop constructs or considering deviations like redoing a task. Thus, an algorithm is needed which alters the input sets for Algo. 1 in case of these previously mentioned scenarios. The trigger for this processing (cf. Algo. 3) is an enactment of task \( j_n \) of the de jure workflow.

To differentiate between individual task instances in case of a repeated enactment, the decision variables in \( T \) are extended with a second index variable \( l \), e.g. \( t(j_n,l) \), denoting the numbering of task instances referencing one single task, here \( j_n \), of the de jure workflow. This second index also simplifies the validation of the constraint Absence \((t_a,n)\) because \( n \) might be compared to the second index \( l \) of the tasks that have already been executed. At first, the variable \( t(k,l) \) corresponding to task \( j_n \) has to be found. The following condition checks whether the enacted task was the first of a loop construct. If so, the CSP is extended considering a possible further execution within the loop. Thus, a new decision variable \( t(j_m,l+1) \), for each task \( j_m \) in the loop, is included with an increased numbering variable \( (l + 1) \). Domains have to be expanded and constraints considering the new loop tasks have to be incorporated.

5 Conclusion and Future Work

In this paper we presented a novel concept for workflow flexibility by deviation that combines procedural and declarative workflow paradigms. Upcoming doc-
Input : Task $j_n$

1. find $t_{(k,l)}$ such that $k = j_n$ and $t_{(k,l)} \in T \setminus F$;
2. if $isFirstTaskInLoop(j_n)$ then
   3. foreach $j_m \in loop$ do
      4. $T = T \cup \{t_{(j_m, l+1)}\}$, with $m = |T|$;
      5. foreach $D_i \in D$ do $D_i = D_i \cup \{|T| - 1\}$;
      6. addConstraints with usual loop constraints;
   7. end
3. end

Algorithm 3: CSP extension

Elements and extracted semantic information are used to determine the current state of the workflow and for control purposes. In order to react to deviations and still propose the best way of proceeding with the workflow, constraint satisfaction problem solving is applied. Future work will focus on a detailed elaboration of the single algorithms and possible improvements to achieve better results. Subsequently, the implementation will be evaluated against related approaches. Additionally, the concept will be developed further, as the workflow designer should be granted more freedom to choose how strict or flexible the constraints should be treated for worklist handling and, furthermore, which and how countermeasures could be specified.

References


Current State and Further Development of a Case-Based Framework for Early Phases of Architectural Conceptual Design

Viktor Ayzenshtadt

University of Hildesheim, Institute of Computer Science
Samelsonplatz 1, 31141 Hildesheim, Germany
ayzenht@uni-hildesheim.de

Abstract

This work provides a summary of the most recent publications in the context of the case-based architectural design support system MetisCBR. It also provides an overview of the features that are currently being developed to extend the functionality of the system.

Keywords: case-based design, retrieval strategies, cognitive model, explanation

1 Introduction

MetisCBR [4] is a case-based framework that was created as a purely CBR-based retrieval engine to support the early phases in architectural conceptual design. The framework’s goal is to increase the efficiency of the design process, i.e., to provide the target user group (architects) with a tool that can find helpful and inspirational designs for the given task (or its sub-tasks). By means of applying the techniques of case-based reasoning [1] (especially the methods of case-based design), an architectural design process can be enriched with data available from experiences made by architects during the creation of previous similar designs (i.e., similarly structured or designed for similar purpose). The framework is currently developed as part of an ongoing PhD work and is used for prototypical implementation of the functionalities for research intentions and of the results of the accompanying studies. The basis of MetisCBR is built on the case-based learning agents paradigm, that is, the agents learn from previous experiences (in our case interpreted as retrieval processes for search of similar architectural designs) in order to apply the best possible strategy or to avoid the application of the strategy with a possible negative outcome. Currently, the framework is in its advanced stage of development where the initial functionality (retrieval and learning of previous queries) will be extended to the process-oriented functionality with retrieval, adaptation, explanation, and extended learning features. The current architecture of MetisCBR is shown in Figure 1.

In this paper, we will present the most recent work published in the context of our research. We will also report which of the features named above we are going to develop next to enhance the functionality of MetisCBR for our further research activities.
Figure 1. The current general structure of MetisCBR. The coordination agent (Coordinator) is responsible for the whole retrieval process: it selects the most suitable retrieval strategy for the current query and creates a retrieval container that conducts the actual retrieval with CBR methods within the case base of previous designs. The container can resolve the complete query or parts of it (the concrete task depends on user criteria), it is also possible to run multiple containers concurrently. Other agents of the system help to transform the query into the language format understandable for agents (GraphML agent), communicate with the user interface (Gateway) or maintain the case base of building designs (Maintainer). This Figure is an adapted version of Figure from [4].

2 Recent Work

Our most recent published work dealt mostly with comparison of MetisCBR to other retrieval engines with identical purpose (i.e., the search for similar architectural designs) and with initial extension to a process-oriented system. In the next sections we describe this recent work.

2.1 Comparative Evaluations

The goal of the evaluations was to examine under which architectural scenarios (i.e., tasks of constructing a building design for a certain purpose, e.g., an apartment for elderly married couple) MetisCBR would perform best and what the current technical boundaries of the system are (e.g., what is the most complex possible query that the system currently can handle).
The first comparative evaluation [5] was conducted between our framework and a retrieval coordination middleware *KSD Coordinator* that has access to the methods of exact subgraph matching and database search. This evaluation has shown that MetisCBR is currently more suitable for scenarios where a sufficient number of cases (i.e., architectural designs) should be found to provide an inspirational space for an architect. However, when it comes to the scenarios where an exact connection within a building design should be detected to take a look at this connection in other context, a subgraph isomorphism method would be a preferable one.

The second evaluation conducted in [7] compared MetisCBR to another exact graph matching approach VF2 [6] and an index-based searching approach for a certain number of queries. It was also aimed to proof if the methods can handle complex queries with big number of rooms and connections. All things considered, MetisCBR was able to be rated as the second best retrieval method (preceded by the VF2 method) and earned the joint first place for the handling of the complex queries (together with the VF2 method).

### 2.2 Initial Extension to a Process-Oriented System

In order to improve the system performance and the quality of results returned by MetisCBR, we decided to extend it to a process-oriented system, where the retrieval strategies will be embedded in the complete conceptual design process, the processes themselves will be categorized and assigned to a user when a certain behavior is detected (i.e., the processes will be seen as *user models*). To provide structure to retrieval strategies and processes/user models, we conducted a study where the target group (architects) played the role of the system and were asked to manually find the most similar case in a case base of printed designs and to model their similarity assessment strategy afterwards. The results of this study allowed us to infer definitions for strategy and process according to architectural requirements for implementation in our system. The complete results and definitions for strategy and process are available in [3].

### 3 Further Development

To enhance the functionality of MetisCBR by conducting a novel research and to gain more interest in the research area of case-based design among the young academia community a number of special student (graduation) projects were recently started to initiate the system’s further development. We decided to initiate research and development activities in the following directions:

- **Retrieval strategies** – implementation of a number of new strategies for the retrieval phase according to the requirements of strategy definition from [3].
- **Cognitive architectures and user models** – a comparison of MetisCBR’s current system architecture with a number of well-known cognitive architectures.
- **Explanations** – conceptualization of explanation patterns for retrieval results.

The short descriptions of each project are provided in the following sections.
3.1 Retrieval Strategies Implementation

The implementation of retrieval strategies according to the definition provided in [3] is an essential part of examination of suitability of the definition for a ‘real-world’ use. Currently, a number of custom new strategies are being implemented as part of a bachelor thesis. An evaluation with participation of a domain expert will show if the strategies provide an improvement for the retrieval phase, i.e., if the new strategies return better results than the old ones. It is then planned that, depending on the evaluation results, these new strategies will replace or complement the currently available strategies. An exemplary new strategy previously published in [3] is provided in Figure 2.

![Figure 2](image_url)

Figure 2. An exemplary strategy from [3] constructed according to the definition provided in this work (Figure from [3]).

3.2 Comparison with Cognitive Architectures

Cognitive architectures such as ACT-R [2] or Clarion [9] are some of the previously developed models of abstract human behavior. These models will be analyzed, configured to correspond to the purpose of MetisCBR and evaluated using predefined criteria. The evaluation, as well as analysis, is part of a master thesis and will show which features can be adapted from these architectures to improve MetisCBR’s user models development.

3.3 Explanation Goals and Patterns

Finally, a non-graduation project has been started that is aimed to explore the explanation goals described in [8] and to adapt them for construction of explanation patterns for MetisCBR w.r.t. these goals. This project is the next
step of addition of an explanation component for results returned during the retrieval phase. This explanation component will help us to improve user’s trust in the system, as it will provide an additional data to the results by enriching them with information about relevance and justification of single results.

4 Conclusion

Concluding, we can summarize that the development of MetisCBR has stepped into its next planned phase, i.e., the system does not conduct retrieval only, but will provide additional functions, such as explanation and personalization of search behavior (user models), followed by adaptation and extended learning features in the upcoming phases.

References

Learning Word Embeddings from Tagging Data: A methodological comparison

Thomas Niebler¹, Luzian Hahn¹, and Andreas Hotho¹,²

¹ Data Mining and Information Retrieval Group, University of Würzburg (Germany)
{nibbler, hotho}@informatik.uni-wuerzburg.de
luzian.hahn@stud-mail.uni-wuerzburg.de
² L3S Research Center Hanover (Germany)

Abstract. The semantics hidden in natural language are an essential building block for a common language understanding needed in areas like NLP or the Semantic Web. Such information is hidden for example in lightweight knowledge representations such as tagging systems and folksonomies. While extracting relatedness from tagging systems shows promising results, the extracted information is often encoded in high dimensional vector representations, which makes relatedness learning or word sense discovery computationally infeasible. In the last few years, methods producing low-dimensional vector representations, so-called word embeddings, have been shown to yield extraordinary structural and semantic features and have been used in many settings. Up to this point, there has been no in-depth exploration of the applicability of word embedding algorithms on tagging data. In this work, we explore different embedding algorithms with regard to their applicability on tagging data and the semantic quality of the produced word embeddings. For this, we use data from three different tagging systems and evaluate the vector representations on several human intuition datasets. To the best of our knowledge, we are the first to generate embeddings from tagging data. Our results encourage the use of word embeddings based on tagging data, as they capture semantic relations between tags better than high-dimensional representations and make learning with tag representations feasible.

1 Introduction

Automatically assessing the degree of semantic relatedness between words, i.e., the relatedness of their actual meanings, in such a way that it fits human intuition is an important task with a variety of applications, such as ontology learning for the Semantic Web [3], tag recommendation [15] or semantic search [13]. Semantic relatedness information of words has been extracted from a variety of sources like plain text [7], website navigation [21, 27] or social metadata [5, 8, 17]. Among others, tagging data from social tagging systems like BibSonomy [4] or Delicious [4] are useful to extract high-quality semantic relatedness information, e.g., for ontology learning [3].

Traditionally, assessing the degree of semantic relatedness between tags utilizes sparse, high-dimensional vector representations of those tags, which are constructed

1 https://www.bibsonomy.org
2 http://www.delicious.com

Copyright © 2017 by the paper’s authors. Copying permitted only for private and academic purposes.
In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
from tag contexts based on posts in social tagging systems [8]. The semantic relatedness can then be estimated using the cosine measure of the corresponding tag vectors [17]. Finally, evaluating the quality of the estimated scores is usually performed by directly correlating them to human intuition [6, 11, 25]. In recent years, many techniques have been proposed to represent words by dense, low-dimensional vectors [20, 23, 28]. These so-called word embeddings have been shown to yield extraordinary structural features [16, 19] and are applied in machine translation or text classification. Furthermore, word embeddings often outperform high-dimensional representations in tasks such as measuring semantic relatedness [1, 16].

Problem Setting. Traditionally, tags are represented by sparse, high-dimensional vectors [8, 26]. However, although Cattuto et al. have shown that tagging data contain meaningful semantics [8], the correlation of semantic relatedness scores from those vectors with human intuition still leaves room for improvement. Furthermore, the high dimensionality of those representations renders many algorithms using them computationally expensive. Up to this point, there have been no extensive attempts to generate word embeddings from social tagging data. All prior studies rely on high dimensional tagging vectors or reduce the vector space arbitrarily by cutting the dimensionality of the space by a fixed number, which in turn decreases the fit of the resulting relatedness scores to human intuition.

Contribution. We contribute a thorough exploration of the applicability and optimization of three well-known embedding algorithms on tagging data. We first analyze the parameters of each algorithm, before we optimize these settings to produce the best possible word embeddings from tagging data. Then, we compare the embeddings of each algorithm with each other as well as with traditional sparse representations by evaluating them on human intuition. We show that all produced embeddings outperform high-dimensional vector representations. We discuss the results in the light of other semantic relatedness approaches and show that we reach competitive results, on par with recent work on extracting semantic relatedness, which opens another high quality source of information for semantic relatedness.

Structure of this work. We first cover the related work in Section 2 and any essential theoretical background of this work in Section 3. Afterwards, we investigate three well-known algorithms with respect to their applicability on tagging data in Section 4. In Section 5, we describe the datasets we used in our experiments. Section 6 outlines our experiments, where we compare all generated vector representations with regard to their semantic content. Section 8 concludes this work.

2 Related Work

The related work to this paper can be roughly put in two groups: Word Embedding algorithms and semantics of tagging data as well as their applications.

Word Embeddings. The concept of word embeddings, i.e., word representations in low dimensional vector spaces dates back at least to 1990, when Deerwester presented LSA [9], which by factorizing a term-document matrix effectively produced a dimension reduction of the term vector space. In 2003, Bengio et al. presented their neural probabilistic language model [2]. The goal of this work was to overcome
the curse of dimensionality and learn distributed word representation in a low-dimensional vector space. However, the wide-spread use of word embeddings only really took off in 2013, when Mikolov et al. presented a similar, yet scalable and fast approach to learn word embeddings [20]. Generally, such methods train a model to predict a word from a given context [2] [20]. Other embedding methods focus on factorizing a term-document matrix [6] [23]. In [1], Baroni et al. showed that all those methods generally exhibit a notably higher correlation with human intuition than the standard high-dimensional vector representations proposed in [26]. There also exist several graph embedding algorithms. The LINE algorithm [28] attempts to preserve the first- and second-order proximity of nodes in their corresponding embedding relations. Perozzi et al. [24] proposed “DeepWalk”, an approach based on random walks on graphs and the subsequent embedding using Word2Vec.

Social Tagging Systems. In [12], Golder and Huberman noted that with increasing use, usage data from social tagging systems exhibited an emerging structure, which was later called a folksonomy [29]. Mika noted that these emerging structures, i.e., folksonomies, could even represent light-weight ontologies [18]. Using the folksonomy structure, it was possible to extract information about semantic relatedness between tags [8] [17]. The evaluation of this semantic relatedness information on human intuition showed that tagging data contain a considerable amount of semantic information, thus enabling further applications of tagging data. Applications of these emerging structures can be found in tag recommendation [15], ontology learning [3] and tag sense discovery algorithms [22].

3 Technical Background

In the following, we will describe the technical background for this paper. First, we define the term folksonomy. Secondly, we introduce how to extract information about semantic relatedness from folksonomies.

Folksonomies. Folksonomies are the data structures emerging from social tagging systems. The term has been coined by Van der Wal in 2005 as a portmanteu of “folks” and “taxonomy” [29]. In these systems, users collect resources and annotate them with freely chosen keywords, so-called tags. Examples are BibSonomy, Delicious, FlickR or last.fm. We follow the definition given by [14]:

A folksonomy is a tuple \((U,T,R,Y)\) of sets \(U, T, R\) and a tripartite relation \(Y \subseteq U \times T \times R\). The sets \(U, T\) and \(R\) represent the sets of users, tags and resources, respectively, while \(Y\) represents the set of tag assignments. A post is the collection of tag assignments with the same user and same resource.

Extracting Semantic Relatedness from Folksonomies. After Golder and Huberman argued that the emerging structure of folksonomies contains considerable semantic information [12], Cattuto et al. proposed a way to extract this information [6]. They used a context-co-occurrence based vector representation for the tags and experimented with different context choices, such as tag-tag-context or tag-resource-context, i.e., all assigned tags of a posted resource by either a specific user or all users. Both of these context choices were shown to estimate human-perceived semantic relatedness better than other context variants. In this work, we generally use the tag-tag-context. The resulting vector representations follow the definition
given in [26] and are based on the co-occurrence counts of tags in their respective contexts. More concretely, a vector representation \( v_i \) of a tag \( t_i \in V \) in a given vocabulary is then a \( |V| \)-dimensional vector, where \( v_{ij} := \#\text{coocc}_{\text{post}}(i, j) \). To finally receive a notion of the degree of semantic relatedness between two tags \( i \) and \( j \), one can compare the corresponding vectors \( v_i \) and \( v_j \) using the cosine measure \[ \text{cossim}(v_i, v_j) := \frac{\langle v_i, v_j \rangle}{\|v_i\| \cdot \|v_j\|} \] [17].

4 Applicability of Embedding Algorithms on Tagging Data

This section describes the different embedding algorithms that we explored. For each algorithm, we give a short summary, enumerate the parameters for each model and shortly discuss how it can be applied to tagging data.

**Word2Vec** The most well-known embedding algorithm used in this work is the Word2Vec algorithm [20], which is actually comprised of two algorithms, SkipGram and CBOW (Cumulative Bag of Words). Word2Vec trains a shallow neural network on word sequences to predict a word from its neighbors in a given context window.

**Parameterization.** Word2Vec takes two parameters. The first parameter is the window size, which determines the amount of neighboring words in a sequence considered as context from which a word will be predicted. The second parameter is the number of negative samples per step. This is done to decrease complexity of solving the proposed model by employing a noise contrastive estimation approach.

**Applicability.** The Word2Vec algorithm normally processes sequential data. However, the sequence of tags normally does not hold any meaning, so this could possibly pose a problem if the window size is chosen too small. In order to be able to apply Word2Vec on tagging data, we grouped the tag assignments into posts and fed the random succession of tags as sentences into the algorithm.

**GloVe** GloVe is an unsupervised learning algorithm for obtaining vector representations for words [23]. Its main objective was to capture semantic relations such as \( \text{king} - \text{man} + \text{woman} \approx \text{queen} \). Training is performed on aggregated global word-word co-occurrence statistics from a corpus.

**Parameterization.** The main parameters of the GloVe algorithm are \( x_{\text{max}} \) and \( \alpha \). \( x_{\text{max}} \) denotes an influence cutoff for frequent tags while \( \alpha \) determines the importance of infrequent tags. According to [23], GloVe worked best for \( x_{\text{max}} = 100 \) and \( \alpha = 0.75 \). We will choose these as initial values in our experiments.

**Applicability.** Since GloVe depends on co-occurrence counts of words in a corpus, it is very easy to apply on tagging data. For this, we construct the tag-tag-context co-occurrence matrix and can then directly feed it into the algorithm.

**LINE** The goal of the LINE embedding algorithm is to create graph embeddings where the first- and second-order proximity of nodes are preserved [28]. The first-order proximity in a network is the local pairwise proximity of two nodes, i.e., the

---

6 In the course of this work, every time we refer to Word2Vec, we talk about the CBOW algorithm, as is recommended by [20] for bigger datasets.
weight of an edge connecting these two nodes. The second-order proximity of two nodes in a network is the similarity between their first-order neighborhoods.

**Parameterization.** LINE takes two different parameters: the amount of edge samples per step and the amount of negative samples per edge. To decrease complexity of solving the proposed model in [28], the authors employed a noise contrastive estimation approach as proposed by [20] using negative sampling. Furthermore, to avoid high edge weights to outweigh lower weights by letting the gradient explode or vanish, LINE employs a sampling process of edges and then ignores their weights instead of actually using the edge weights in its objective function.

**Applicability.** Similar to GloVe, this algorithm processes a network with weighted edges, such as a co-occurrence network. Thus, we only have to construct the co-occurrence network from the tagging data and apply LINE on that network.

**Common Parameters** While each of the mentioned algorithms can be tuned with a set of different parameters, they have some parameters in common. First, the embedding dimension determines the size of the produced vectors. A higher embedding dimension allows for more degrees of freedom in the expressiveness of the vector, i.e., it can encode more information about word relations. Standard ranges for embedding dimensions are between 25 and 300. Secondly, the initial learning rate of an algorithm determines its convergence speed. Fine-tuning that parameter is crucial to receive optimal results, because if chosen badly, the learning process either converges very slowly or might be unable to converge at all.

## 5 Datasets

In this work we use two different kinds of datasets to evaluate embedding algorithms on tagging data. That is, the actual tagging datasets which provide tagging metadata and human intuition datasets (HIDs) which we employ to evaluate semantic relatedness. In the following we first describe three datasets containing tagging data from which we later derive tag embeddings. Then we introduce all human intuition datasets containing human-assigned scores of similarities to word pairs.

### 5.1 Tagging Datasets to Derive Word Embeddings

We study datasets of three public social tagging systems. In order to ensure a minimum level of commonly accepted meaning of all tags, each dataset is restricted to the top 10k tags. Additionally, we only considered tags from users who have tagged at least 5 resources and resources which have been used at least 10 times. We also removed all invalid tags, e.g., containing whitespaces or unreadable symbols.

**BibSonomy.** The social tagging system BibSonomy provides users with the possibility to collect bookmarks (links to websites) or references to scientific publications and annotate them with tags [4]. We use a freely available dump of BibSonomy, covering all tagging data from 2006 till the end of 2015. After filtering, it contains 9,302 distinct tags, assigned by 3,270 users to 49,654 resources in 630,962 assignments.

[7] [http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/](http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/)
Delicious. Like BibSonomy, Delicious is a social tagging system, where users can share their bookmarks and annotate them with tags. We use a freely available dataset from 2011 [30]. Delicious has been one of the biggest adopters of the tagging paradigm and due to its audience, contains tags about design and technical topics. After filtering, the Delicious dataset contains 10,000 tags, which were assigned by 1,685,506 users to 11,486,080 resources in 626,690,002 assignments.

CiteULike. We took a snapshot of the official CiteULike page from September 2016. Since CiteULike describes itself as a “free service for managing and discovering scholarly references”, it contains tags mostly centered around research topics. After filtering, the CiteULike dataset contains 10,000 tags, which were assigned by 141,395 users to 4,548,376 resources in 15,988,259 assignments.

5.2 Human Intuition Datasets (HIDs)

As a gold standard for semantic relatedness as it is perceived by humans, we use several datasets with human-generated relatedness scores for word pairs. In the following, we will describe all of the used datasets briefly.

WS-353. The WordSimilarity-353 dataset consists of 353 pairs of English words and names [10]. Each pair was assigned a relatedness value between 0.0 (no relation) and 10.0 (identical meaning) by 16 raters, denoting the assumed common sense semantic relatedness between two words. Finally, the total rating per pair was calculated as the mean of each of the 16 users’ ratings. This way, WS-353 provides a valuable evaluation base for comparing our concept relatedness scores to an established human generated and validated collection of word pairs.

MEN. The MEN Test Collection [6] contains 3,000 word pairs together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk [11]. Contrary to WS-353, the similarity judgments are relative rather than absolute. Raters were given two pairs of words at a time and were asked to choose the pair of words that was more similar. The score of the chosen pair, i.e., the pair of words that was more similar, was then increased by one. Each pair was rated 50 times, which leads to a score between 0 and 50 for each pair.

Bib100. The Bib100 dataset has been created in order to provide a more fitting vocabulary for the research and computer science oriented tagging data that we investigate [12]. It consists of 122 words in 100 pairs, which were judged 26 times each for semantic relatedness using scores from 0 (no similarity) to 10 (full similarity).

MTurk. In [25], Radinsky et al. created an evaluation dataset specifically for news texts [13]. We use this dataset as a topically remote evaluation baseline in order to get a notion how intrinsic semantic relations are captured by both the tagging data and the generated embeddings. The dataset at hand consists of 287 word pairs and 499 words. 23 humans judged relatedness on a scale from 1 (unrelated) to 5 (related).

\[\text{http://www.zubiaga.org/datasets/socialbm0311/}\]
\[\text{http://www.citeulike.org/tag/data.adp}\]
\[\text{http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html}\]
\[\text{http://clic.cimec.unitn.it/~elia.bruni/MEN}\]
\[\text{http://www.dmir.org/datasets/bib100}\]
\[\text{http://www.kiraradinsky.com/Datasets.html}\]
6 Experimental Setup and Results

In the following, we describe the conducted experiments and present the results for each experiment. Due to space limitations, we only report results for MEN.  

6.1 Preliminaries

Evaluating Word Vector Representations. Very often, the quality of semantic relatedness encoded in word vectors is assessed by how well it fits human intuition. Human intuition is collected in HIDs as introduced in Section 5. The most widely-used method to evaluate semantic relatedness on such datasets is to compare human scores of the semantic relatedness between two words with the cosine similarity scores of the corresponding word vectors. The comparison is done by calculating the Spearman rank correlation coefficient \( \rho \), which compares two ranked lists of word pairs induced by the human relatedness scores and the cosine scores [1, 23].

Baseline: Tag-Tag-Context Co-Occurrence Vectors. As a baseline, we produced high dimensional co-occurrence counting vectors from all three tagging datasets. As described in Section 3, co-occurrence of tags was counted in a tag-tag-context, i.e., the context of a tag was given as the other tags annotated to a given resource by a certain user [8]. Since there is no option to vary the dimension of the tag-tag-context co-occurrence vectors except truncating the vocabulary, we only report the values for a truncated vocabulary of 10,000 tags in Table 1. Still, we give all of the reported results as baselines in the subsequent figures.

Parameter Settings. For each of the following algorithms, we conducted the experiments as follows: As initial parameter setting, we used the standard settings that come with the implementation of each algorithm. The corresponding values are given in Table 2. We then varied the initial learning rate for each algorithm in the range of 0.01 to 0.1 in steps of 0.01. After that, we varied the embedding dimension on the set of \{10, 30, 50, 80, 100, 120, 150, 200\}. For Word2Vec and LINE, we now varied the number of negative samples on the set of \{2, 5, 8, 12, 15, 20\}. For GloVe, we varied \( x_{\text{max}} \in \{25, 50, \ldots, 200\} \) and \( \alpha \in \{0.5, 0.55, \ldots, 1\} \) simultaneously. Finally, for Word2Vec, we varied the context window size between \{1, 3, 5, 8, 10, 13, 16, 20\}, while for LINE, we varied the number of samples per step on \{1, 10, 100, 1000, 10000\} · 10\(^6\). To rule out influence of a random embedding initialization, each experiment was performed 10 times and the mean result was reported. After each experiment, we chose the best performing parameter settings on the respective tagging datasets across the four evaluation datasets and used them for all other experiments.

6.2 Embedding Evaluation Results

We will now present the evaluation results. For each algorithm, Table 3 gives the parameter settings which produced the highest-scoring embeddings. In each figure, we report both the evaluation results of the embeddings for a given parameter as well as the corresponding baselines produced by the high-dimensional vector representations given in Table 1.

\footnote{All result figures are publicly available at http://www.dmir.org/tagembeddings}
Table 1: Spearman correlation values for the co-occurrence baseline. For all evaluation datasets, we gave the total number of pairs in the original dataset and the number of matched pairs, i.e., where both words were present in the tagging corpus.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BibSonomy</th>
<th>Delicious</th>
<th>CiteULike</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-353 (353)</td>
<td>0.433 (158)</td>
<td>0.486 (202)</td>
<td>0.186 (139)</td>
</tr>
<tr>
<td>MEN (3000)</td>
<td>0.415 (463)</td>
<td>0.577 (1376)</td>
<td>0.423 (404)</td>
</tr>
<tr>
<td>MTurk (287)</td>
<td>0.604 (62)</td>
<td>0.508 (103)</td>
<td>0.469 (53)</td>
</tr>
<tr>
<td>Bib100 (100)</td>
<td>0.623 (100)</td>
<td>0.632 (94)</td>
<td>0.270 (87)</td>
</tr>
</tbody>
</table>

Table 2: Initial parameter values for each algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>initial learning rate</th>
<th>dimension</th>
<th>samples per step</th>
<th>negative samples $$(x_{max}, \alpha)$$</th>
<th>window size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE</td>
<td>0.025</td>
<td>100</td>
<td>$100 \cdot 10^6$</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>GloVe</td>
<td>0.05</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>(100, 0.75)</td>
</tr>
<tr>
<td>Word2vec</td>
<td>0.025</td>
<td>100</td>
<td>-</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3: Best parameter values for each algorithm for the MEN dataset on Delicious data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>initial learning rate</th>
<th>dimension</th>
<th>samples per step</th>
<th>negative samples $$(x_{max}, \alpha)$$</th>
<th>window size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE</td>
<td>0.1</td>
<td>100</td>
<td>$100 \cdot 10^6$</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>GloVe</td>
<td>0.1</td>
<td>120</td>
<td>-</td>
<td>(100, 0.75)</td>
<td>-</td>
</tr>
<tr>
<td>Word2vec</td>
<td>0.1</td>
<td>100</td>
<td>-</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>

Word2Vec. Although Word2Vec is meant to be applied on sequential data, as opposed to the bag-of-words nature when assigning tags, the generated embeddings yielded better correlation scores with human intuition than their high-dimensional counterparts. However, we did not shuffle the tag sequence in posts, which is left to future work. Figure 1a shows that fine-tuning the initial learning rate exhibits a great effect on the quality of word embeddings from BibSonomy, with general peak performance at $\alpha = 0.1$, while Delicious data seem unaffected. Increasing the embedding dimension only improves the embeddings’ semantic content up to a certain point, which is mostly reached at around a very low number of dimensions between 30 and 50 (Figure 1b). Anything above does not notably increase performance of the embeddings. The number of negative samples seems sufficiently high at 10 samples and even earlier for Delicious and CiteULike (Figure 1c). The context size had negligible impact on the semantic content of the generated embeddings (Figure 1d).

GloVe. GloVe generates embeddings from co-occurrence data. As mentioned in Section 4, GloVe is parameterized by the learning rate, the dimension of the generated embeddings as well as by the weighting parameters $x_{max}$ and $\alpha$, which regulate the importance of low-frequency co-occurrences in the training process. While the learning rate does not show a great effect on embeddings generated from Delicious data, fine-tuning influences the semantic content of CiteULike and BibSonomy embeddings notably (Figure 3a). Mostly, peak performance is reached at an embedding dimension of 100 or even earlier, except for Delicious (Figure 3b). Furthermore, BibSonomy is quite sensitive to poor choices of $x_{max}$ and $\alpha$, i.e., if chosen too high, performance suffers greatly (Figure 3c). Delicious and CiteULike seem unaffected by those parameters, at least in our experimental ranges (Figures 3d and 3e).

LINE. LINE generates vertex embeddings from graph data, preserving the first- and second-order proximity between vertices. Its parameters are the initial learning
rate, the embedding dimension, the number of negative samples per edge and the number of samples per training step. While influence of the initial learning rate is visible, it is not as great as with GloVe (cf. Figure 2a). Also, the embedding dimension gives similar results above 50 and only lets performance suffer if chosen too small (cf. Figure 2b). Interestingly enough, Figure 2c shows that the number of negative samples seems to have almost no effect on the generated embeddings across all tagging datasets. In contrast, choosing the number of samples per step exerts great influence of the resulting embeddings, as can be seen in Figure 2d.

7 Discussion

Across all algorithms, fine-tuning the initial learning rate greatly improves results for embeddings based on BibSonomy, especially with GloVe. The effect of the embedding dimension is much less pronounced across all three embedding algorithms. Peak evaluation performance is often reached with an embedding dimension between 50 and 100 and stays quite stable with increasing dimension. Varying the number of negative samples influences evaluation results of BibSonomy, but only at a very high number of 20 negative samples. In contrast, Delicious and CiteULike only show small performance changes already with 3 to 5 samples. Finally, GloVe’s weighting factors $x_{\text{max}}$ and $\alpha$ negatively influence results on BibSonomy, while barely affecting evaluation performance on Delicious and CiteULike, due to BibSonomy being our smallest tagging dataset with rarely any co-occurrences above a high $x_{\text{max}}$

Generally, all investigated embedding algorithms produce high-quality embeddings from tagging data. Although [8] found that tagging data contain high-quality semantic information, the high-dimensional vector representation proposed there
Figure 2: Evaluation results for embeddings generated by LINE on the MEN dataset. While the initial learning rate, the embedding dimension and the amount of samples per step exert a notable influence on the evaluation result, increasing the number of negative samples per edge only slightly improves results.

seems to not optimally capture this information when evaluated on human judgment (see Table 1). In contrast, the generated embeddings seem better suited to capture that information, as they mostly outperform the tag-tag-context based co-occurrence count vectors (Section 5). Furthermore, the best result achievable on WS-353 in this work is from Delicious data using the GloVe algorithm of around 0.7 (cf. Figure 4b), which is on par with other well-known works, such as ESA [11], which is based on Wikipedia text, achieving correlation around 0.748, or the work done by Singer et al. on Wikipedia navigation [27] with the highest correlation at 0.76, but generally achieving scores around 0.71.

8 Conclusion

In this work, we explored embedding methods and their applicability on tagging data. We conducted parameter studies for three well-known embedding algorithms in order to achieve the best possible embeddings based on tagging data regarding their fit to human intuition of semantic relatedness. Our results indicate that i) tagging data provide a viable source to generate high-quality semantic embeddings, even on par with current state-of-the-art methods and ii) that in order to achieve competitive results, it is necessary to choose correct parameters for each algorithm instead of the standard parameters. Overall we bridged the gap between the fact that tagging data yield considerable semantic content and the current state-of-the-art methods to produce high-quality and low-dimensional word embeddings. We expect our results to be of special interest for folksonomy engineers and others working with semantics of tagging data. Future work includes investigation of the influence of different vector representations on tagging-based real-world applications, such as
Figure 3: Evaluation results for embeddings generated by GloVe on the MEN dataset. The initial learning rate only influences the smaller tagging datasets, while Delicious profits most from increasing dimension. BibSonomy is influenced by a high cutoff $x_{\text{max}}$ the most.

Figure 4: Evaluation results for embeddings generated by the best parameter settings across the different tagging datasets. GloVe mostly produces the best embeddings and is the only algorithm to always outperform the baseline.

tag recommendations in social tagging systems, tag sense discovery and ontology learning algorithms. Furthermore, we want to try to improve the fit of tagging embeddings to human intuition by applying metric learning approaches or alignment approaches to external knowledge bases, e.g., WordNet or DBPedia.

References

Math Object Identifiers – Towards Research Data in Mathematics

Michael Kohlhase
Computer Science, FAU Erlangen-Nürnberg

Abstract. We propose to develop a system of “Math Object Identifiers” (MOIs: digital object identifiers for mathematical concepts, objects, and models) and a process of registering them. These envisioned MOIs constitute a very lightweight form of semantic annotation that can support many knowledge-based workflows in mathematics, e.g. classification of articles via the objects mentioned or object-based search. In essence MOIs are an enabling technology for Linked Open Data for mathematics and thus makes (parts of) the mathematical literature into mathematical research data.

1 Introduction

The last years have seen a surge in interest in scaling computer support in scientific research by preserving, making accessible, and managing research data. For most subjects, research data consist in measurement or simulation data about the objects of study, ranging from subatomic particles via weather systems to galaxy clusters.

Mathematics has largely been left untouched by this trend, since the objects of study – mathematical concepts, objects, and models – are by and large abstract and their properties and relations apply whole classes of objects. There are some exceptions to this, concrete integer sequences, finite groups, or $\ell$-functions and modular form are collected and catalogued in mathematical data bases like the OEIS (Online Encyclopedia of Integer Sequences) [Inc; Slo12], the GAP Group libraries [GAP, Chap. 50], or the LMFDB ($\ell$-Functions and Modular Forms Data Base) [LMFDB; Cre16].

Abstract mathematical structures like groups, manifolds, or probability distributions can formalized – usually by definitions – in logical systems, and their relations expressed in form of theorems which can be proved in the logical systems as well. Today there are about half-a-dozen libraries with $\sim 10^5$ formalized definitions, theorems, and proofs; e.g. the libraries of Mizar [Miz], Coq [Tea], and the HOL systems [Har96]. These include deep theorems such as the Odd-Order Theorem [Gon+13] or the Kepler Conjecture [Hal+15].
Finally, the mathematical literature is systematically collected in the two mathematical abstracting services Zentralblatt Math [ZBM] and Math Reviews [AMS], which also (jointly) classify more than 120,000 articles in the Mathematics Subject Classification (MSC) [MSC10] annually.

Unfortunately, while all of these individually constitute steps into the direction of research data, they attack the problem at different levels (object, vs. document level) and direction (description- vs. classification-based) and are mutually incompatible and not-interlinked/aligned systematically.

Finally, only the abstracting systems manage to keep up with the extent (3.5 M articles since 1860) and growth (120,000 articles annually) of the mathematical literature. As a consequence they constitute the only full-coverage information systems for mathematical knowledge. Unfortunately, the MSC classification alone does not give sufficiently focused access to the literature – the time where a working mathematician could “subscribe” to a dozen MSC classes and stay up to date by scanning/reading all articles in them is over: many of the almost 8000 MSCs have more than a hundred articles appearing annually. Incidentally, searches in the zbMATH and MathSciNet databases are mostly by bibliographic metadata, such as authors, years, and keywords – not by MSC classes or the mathematical concepts, objects, or models the user is interested in. Even the new formula search tab in zbMATH [Koh+13], which does give access to formulae does not help much in this situation, since the concept of formula search and the query interface is unfamiliar to most users.

Experience from other scientific fields show us that this intuition that research data and information systems should be about the objects of science is adequate, and indeed why a large infrastructure around these has sprung up. Unfortunately, full formalization in logics and even partial/flexible formalization [Koh13] as developed by the author’s research group that would enable that is currently intractable in practice. In this situation, we propose to develop and deploy an open, and community-based system mathematical object identifiers (MOIs), i.e. handles on mathematical objects that allow to uniquely reference mathematical concepts, objects, and models. Such a system of MOIs (and a central information system that exposes them to the user) would simplify many mathematical information gathering and knowledge management tasks.

In the next section we discuss scientific referencing systems that use object or document handles. In Section 3 we develop a concrete proposal for registering math object identifiers. In Section 5 we sketch some applications of MOIs that justify our claim of information simplification. Section 6 concludes the paper.

2 Handle Systems for Scientific Documents and Objects

Document Object Identifiers [DOI] are persistent digital identifiers for objects (any object really), but are predominantly used for media (normally documents of some kind). DOIs are registered for virtually all scientific publications nowadays and allow location-independent and persistent referencing of the scientific literature and a host of knowledge management applications on top of this. Ser-
vices like the DOI catalog at https://doi.org allow to look publications by their DOI, and the bibliometry and citation analysis profit from the unique handle system.

But (scientific) publications are only a means to the end of conveying information and knowledge about the objects of science. Thus a finer-grained handle system for the objects of science would benefit scientific knowledge management and information retrieval: Publications could be annotated by the objects they talk about and could be indexed by the handles. Indeed, some sciences have developed such handle systems: The most prominent are probably the InChI (IUPAC International Chemical Identifier) [InChi] system in chemistry, which allows to reference any substance by a 27-character string and the CAS registry numbers (CASRN) [CAS] in chemistry. Both systems are widely deployed and are used for referencing and knowledge management (both commercial and public).

These systems profit from the fact that chemistry has a fixed domain – all substances can be built compositionally from a finite set of elements – and the community has developed a standardized way to describe substances by their chemical formulae (the IUPAC nomenclature) that has been essentially fixed for a century now. The InChIs are directly computed from the formula and associated information.

Mathematics works differently, instead of an external domain fixed by nature, mathematics studies the domain of all objects that can be rigorously described by axiomatizations, definitions, and theorems. The method of scientific investigation is that of proof rather than experimentation, modeling, and simulation. Crucially, there is no standardized nomenclature for mathematical objects, only a weakly standardized language of making descriptions. In particular, different descriptions may refer to the same mathematical object – we call them equivalent – and there is no intrinsic way of preferring any of them, since all of them add different insights to understanding this object.

At times, equivalence of descriptions is immediate, in other cases can require deep insights and decades of work to establish. Analogously it may be very hard to tell mathematical objects apart, an extreme example is the two integer sequences \[ \left\lfloor \frac{2^n \log(2)}{\log(2)} \right\rfloor \text{ and } \left\lceil \frac{2^n}{2^{\lceil n/2 \rceil}} \right\rceil \] that match for 777451915729367 terms but are not equal [Slo12]. Another one is the P versus NP problem, where currently do not know whether the classes of problems that can be answered (P) or checked (NP) in polynomial time are equal or not. Indeed, this question is one of the most prominent unsolved problems in theoretical Computer Science.

Therefore it seems intractable to give handles to the “platonic mathematical objects” – even though they have subjective reality to mathematicians – and develop a system of handles for published descriptions instead and treat mathematical equivalence – i.e. if two descriptions refer to the same platonic mathematical object – as a metadata relation that can be added over time.

As there cannot be a normative nomenclature for mathematical objects, we propose to model the math object identifiers after the CAS registry numbers rather than the InChIs from chemistry.
3 The MOI Proposal

I propose that the mathematical community – possibly together with a consortium of scientific publishers – establishes a MOI Registry Organization and process that registers MOIs and provides an open information system about them. This organization and service could be hosted by the newly founded International Mathematical Knowledge Trust (IMKT) [IMKT], the mathematical abstracting services, the IMU [IMU], or even the OpenMath Society [OM].

Like the CAS Registry, the MOI registry should assign numbers called math object numbers (MONs) in sequential order\(^1\) to (sufficiently unique descriptions of) mathematical objects identified by the mathematical community for inclusion in the database. From these numbers the MOI registry derives Math Object Identifiers (MOIs): unique string identifiers (hashes) that may contain check digits or related safety measures.

In principle any persistent, peer-reviewed description of a mathematical concept, object, or model is eligible for registration as long as it is sufficiently different from already registered ones. We restrict ourselves to mathematical objects whose descriptions have been published in some kind of persistent, peer-reviewed medium for three reasons:

i) the set of mathematical objects, even the named ones is infinite (it includes the natural numbers,

ii) we want to select out the set of objects that are of interest to the mathematical community – interesting enough to have passed peer review, and

iii) unless they have been published in a persistent – i.e. immutable – medium, we cannot guarantee that MOIs form persistent references.

Upon registration of a MOI, the Registry establishes a Math Object Record (MOR), which contains contains the mandatory fields

1. the math object number (MON),
2. the registration date (MORDate)
3. a normative reference to the media fragment that constitutes the MO description – we call this a math object description reference (MODRef).

Other metadata can be added directly in the MOR or by linked-open-data methods, but is not considered normative. We will give examples in the next section.

The exact criteria for ensuring peer-review and persistence – the latter may well include versioned MODRefs into versioned media – should be determined by the MOI registry organization.

4 MOI Examples

Examples of eligible descriptions include theorems, definitions, proofs, and examples, published in mathematical journals (see Figure 1), objects in (informal)

\(^1\) Crucially, MONs should not be construed to carry mathematical information. In particular the MONs should be independent of any classification system of mathematical objects such as the MSC or other systematic schemes.
mathematical databases like the OEIS or the LMFDB (see Figure 2), formalizations in a theorem prover library (Figure 3) or locally described MOs (Figure 4). We will now go through these in more detail to build up our intuition about the issues involved.

4.1 DOI-based Math Object Records

The most important class of mathematical concepts, objects, and models is established by publishing (about) them in mathematical articles. Figure 4.1 shows three math object records from an article published by Gregor Fels and Wilhelm Kaup in Acta Math in 2008.

---

**Definition 3.4**

For every real-analytic submanifold $F \subset V$, every $a \in F$ and every $r \in \mathbb{N}$, put

(i) $K_0^a F := T_a F$, and define

(ii) $K_r^{a+1} F$ to be the space of all vectors $v \in K_r^a F$ such that there is a smooth mapping $f : V \to V$ with $f'(a)(T_a F) \subset K_r^a F$, $f(a) = v$ and $f(x) \in K_r^a F$ for all $x \in F$.

**Corollary 3.6**

Suppose that $\dim F \geq 2$ and $K_2^x F = \mathbb{R}^2$ for all $x \in F$. Then $F$ is affinely 2-nondegenerate at every point.

**Proof.** The map $f = \text{Id}$ has the property that $f'(x) T_a F \in K_r^a F$ for every $x \in F$. Hence, the relation $f'(x)(T_a F) = T_a F \not\subset K_r^a F$ implies that $x \not\in K_r^2 F$ and thus $K_r^2 F = 0$ as well as $x \neq 0$. In particular, $F$ is uniformly degenerate.

---

Fig. 1. DOI-based Math Object Records
As specified above, the MORs contain the math object number (MON), the MOI registration date, and a math object description reference (MODRef) – the mandatory data above the first dashed line – as well as additional optional information, such as the object type, a link to an online version of the article, and an image of the referenced description.

Like Uniform Resource Locator-based references, the MODRefs consist of a document reference (here a DOI) and a fragment identifier separated by the # character. Only that due to the nature of the target document – at worst a printed document, on average a PDF, and at best a structured XML document – we have to allow for rigorous references that are not directly machine-actionable. Here we have a PDF and use human-readable reference to a numbered statement, alternatives would be to use references based on page/line/column numbers. Note that the first MOR contains a definition of an object ($K_{r,F}$), which is not given name in the article. Nonetheless, we can give it a MON, and thus make if uniquely referencible. The other two examples are similar in structure, but concern a theorem and a proof.

The “additional information” in the MORRecords in Figure 1 is mostly such that would make a MOR information system more useful. Such a system would allow to search the MOI database, there the “snippet” would be useful, since it allow the user to determine the nature of the object identified by the MOI. Other information would be to record where the MOI is “used” and a list of MOIs used in the fragment. E.g. having the MOIs in the definiendum, would directly give us a (conceptual) dependency relation on MOIs.

Note that MO descriptions need not be fully formal, but they should be sufficiently rigorous to be useful (for human readers). For instance the ones in Figure 1 are clearly informal, but rigorous.

4.2 Archive-Based MOIs

The next large group of mathematical objects are ones collected and published in mathematical archives and libraries. Figure 2 shows two examples, one from the Online Encyclopedia of Integer Sequences (OEIS) and the second from the LMFDB, the database of L-functions, modular forms, and related objects. Here, the MO References consist of the library name and the internal identifier there. Given that most mathematical libraries are web-accessible nowadays, these directly correspond to a URL or URL reference (given via the MOURL here).

Note that the MO descriptions in these mathematical data bases are informal, e.g. all the integer sequences in the OEIS (see the first example in Figure 2): sequences are given by their initial segment (usually around 50 elements if available) and further described by metadata about publications, names, implementations, and even generating functions. The latter have the potential for formality, but are given in an informal and ambiguous ASCII representation – see [LK16] for details.

In the informally based MOIs (from the literature or math data bases in Figures 1 and 2), we have only glossed the MODRefs, so that they can be understood by humans. The MO Registry Organization will need to standardize
Fig. 2. MOR based on informal Archives

machine-actionable fragment references for MORefs that target digital objects (in PDFs, scans, or retro-digitized documents). We envision that this will be an open ended list of MORef types specified in separate standards documents developed when the need arises.

In libraries of formal objects, e.g. in theorem prover libraries we can make use of the uniform referencing systems by MMT URIs developed by the author’s research group. Figure 3 shows an example: MMT URIs are triples consisting of a URI (the namespace of the library), the theory specifier, and a symbol path – all separated by the ? operator. We have transformed the libraries of more than a dozen theorem provers into the OMDoc/MMT format and made them accessible in the MathHub system [MH; Ian+14] via their MMT URI; see [Müll+17] for details. These MMT URIs could become a basis for a MORef standard to be developed by the MOI registry organization.

Fig. 3. MOR from Theorem Prover Libraries

4.3 Locally Defined Math Objects

Finally, we could even think of allowing “locally defined” MOIs, where the MOD-Ref is a full description of the MOI in question (see Figure 4); but this would necessitate the standardization of a suitable representation language. The Open-Math Standard [Bus+04] would be a candidate, but we would need to restrict ourselves to the openmath dialect with persistent content dictionaries, e.g. the core OpenMath CDs [OMCD].
4.4 MOI Non-Examples and Corner Cases

In contrast to all the cases above, mathematical objects in software systems, like e.g. `numpy:mandelbrot` should not be registered as MOIs themselves, since they are not persistent (e.g. between software versions). Instead they can be linked to MOIs, for instance the numpy manual could use the MOI of the Mandelbrot set as part of the description of the respective function. A sufficiently exact published description – e.g. in [Bre12, edition 1] which could give rise to a MOI.

Another example of a legitimate mathematical object that should not (systematically) be registered is “the 4711th zero of the $\zeta$ function” – there are just too many zeros. Unless of course the 4711th one is of particular mathematical interest, but then we would expect it to be discussed in an article, which would warrant a DOI-based MOI as shown above.

An interesting corner case is constituted by objects which do not exist (platonically), such as “the greatest prime”. Such “objects” routinely occur in proofs by contradiction, and sometimes interesting mathematical concepts are discussed, conjectured about, and even subject to theorems and proofs long before they are sufficiently well-defined. The “Euler characteristic” of a polyhedron is a point in case, where the story is a long and protracted one, mostly about “objects that do not exist (as it turns out)” [Lak76].

5 MOI Applications and Management

Given a sufficiently large set of registered MOIs and MOI-based information systems that allow to mathematical practitioners to identify and access MOIs many knowledge management services become possible

1. **Document disambiguation and wikification**: If text fragments are annotated with their MOI, then we can add links to their MO record or to their MORRef target. This allows the reader to disambiguate otherwise context-dependent usages.
2. **MOI-based search engines**: find documents by the objects they talk about
3. *Linked Open Data for Maths*: if the objects are referenced using MOIs in papers, then this data can be spidered and papers can be cross-referenced by objects.

4. *Mathematical concordances*: we can register MOIs for all the non-equivalent variants of a mathematical object – e.g. elementary functions with different branch cuts [Eng+13] – and link other objects (e.g. implementations) to the respective MOI.

The last application is somewhat characteristic of the change MOIs bring to knowledge management in mathematics: many workflows become star-shaped. Whereas a concordance between $n$ systems, needs contributors to be specialists in at least two idiosyncratic systems – better in all $n$, in a MOI-based world, contributors only need to know their own system and be able to find/identify the MOIs; a much more likely skillset.

We see that MOIs are an enabling technology/resource! But we have to acknowledge that there is a chicken-and-egg problem with MOI registration, just as with all other semantic technologies: even though the joining costs are smaller than for other semantic technologies, there is no incentive to do it until there is already a large network of MOIs to connect to. An additional problem of the proposal above is that the basic MOI registration process relies on human mathematicians, which creates a bottleneck on the curation side as well.

Both of these can possibly be alleviated by integrating MOI registration with the scientific publication and review process: When an author submits a paper for publication, she also submits a list of MOI registration candidates which can then be reviewed (and iterated) and approved by the reviewers and submitted by the publishers. This process could be supported by suitably extended \LaTeX macros and environments. For instance the theorem environments from the \texttt{amsthm} package could be extended, so that the author can add a macro like $\texttt{\textbackslash moi\{foo\}}$ to the environment, from which the publisher can first generate a MOI registry application and later semantic annotations in electronic versions of the eventual publication (once the MOI is reviewed and registered).

This “organizational solution” which minimizes problems by putting them into a context where the information is “at our fingertips” can be replaced or supported by methods from linked-open-data, computational linguistics, or AI/machine learning.

6 Conclusion & Future Work

We have proposed a handle-based system of “math object identifiers” to simplify and scale the management of mathematical knowledge and enhance systems that treat mathematical knowledge as research data. We have sketched a few applications that show the usefulness of such a resource. The next step will be to find a host organization for a MOI registry, to implement a prototype MOI information system, and build some of the applications in Section 5 to show the benefits of the proposal. The latter is feasible, since we can bootstrap the MOI
system with archive-based MOIs from the OEIS and LMFDB, as well as formal MOIs from the theorem prover libraries in the MathHub system. Furthermore, we can experiment with informal MORefs from the NNEXUS system [GC14].

Acknowledgements This work has been partially funded by DFG under Grant KO 2428/13-1. The author gratefully acknowledges discussions with the KWARC group and the IMU GDML working group.

References


[CAS] CAS REGISTRY - The gold standard for chemical substance information. URL: http://www.cas.org/content/chemical-substances (visited on 06/25/2017).


[ZBM] zbMATH the first resource in mathematics. URL: http://zbmath.org (visited on 07/02/2017).
Searching an Appropriate Journal for your Paper – an Approach Inspired by Expert Search and Data Fusion

Andreas Henrich\textsuperscript{1} and Markus Wegmann\textsuperscript{2}

\textsuperscript{1}Media Informatics Group, University of Bamberg, Bamberg, Germany
\texttt{andreas.henrich@uni-bamberg.de}
\textsuperscript{2}markus.wegmann@live.de

Abstract. On an abstract level, one is often confronted with some type of classification problem where we have one example instance or a textual query and we are looking for the class most appropriate for this instance or query. More concretely, we consider journals as classes and the papers published in certain journals as constituting and describing the respective class. In this scenario two information needs are conceivable: (1) We know one paper and we are looking for all journals which could potentially contain similar work. (2) We want to write a paper, have a first working title, and are looking for journals which could be potential targets for a submission of that paper. In this work, we transfer methods used in expert search and data fusion to find appropriate journals: Using a flat, title based search query for articles we examine voting models used in expertise retrieval with its different data fusion techniques to find and rank journals associated with the matching articles that potentially contain most suitable other articles. To evaluate the ranking of found journals, we remove several test articles from the applied collection and utilize them as request items with their titles. We assume that—on average—the journals where these test articles have been published should be among the top ranked journals to provide a suitable result. This fully automated evaluation provides the opportunity to execute a huge number of requests against the collection of articles and to evaluate the different voting techniques transferred from expert search.

Keywords: Expert Search, Expertise Retrieval, IR Systems, Collection Search

1 Motivation and Related Work

One of the approaches introduced for expertise retrieval \cite{1} is based on relevant documents retrieved by a search query. These documents vote for their associated authors as candidate experts. In this work, we transfer this approach to another domain:

Our keyword search on a bibliographic collection yields matching articles which vote for the journals where the articles have been published as beneficial sources or targets. Our aim is to identify a technique, which yields and ranks journals that potentially contain other publications and resources which match the information need of the user. This information need targets journals rather than single articles.
In a first step, we don't evaluate the discussed techniques using manually created test data or test users. Instead, we use article-titles from the collection itself to automatically send these titles as search requests. Since we have the information in which journal a single article has been published, we can measure the position of this respective journal in the result ranking and evaluate the algorithms.

This work is based on research in data fusion techniques and their application in the field of expertise retrieval. Different approaches show, that combining multiple retrieval results using voting models can improve retrieval effectiveness [2]. In their survey, Balog et al. [3] present different approaches used in expertise retrieval including the document based voting model. Rank- and score-based fusion techniques are listed and evaluated, mostly based on the work of MacDonald et al. [1]. Furthermore, normalization methods are applied for the underlying candidate expert profiles to gain better results. In the mentioned works, it becomes quite significant that the documents in the upper ranks together with their score values have a disproportionately high impact on the quality of the fusion results; exponential variants of fusion techniques can have better results and prove this fact [3].

In the paper at hand, we investigate how such approaches perform in our setting. We present first promising experimental results and discuss potential future research directions.

It should be mentioned that existing journal recommenders from publishers—like EndNote’s manuscript matcher, Elsevier’s journal finder, or Springer’s journal suggester—are obviously related to our approach. However, these systems apply complex ranking schemes using much more information than our simple approach discussed in this short paper. The aim of our paper is to investigate the capability of rather simple voting techniques in the sketched scenario. A comparison with the existing journal recommenders will be an interesting next step but is out of scope for this short paper.

2 Applied Ranking Techniques

This section describes the utilized ranking techniques. For the flat, title-based article search—i.e. the underlying document ranking—we use Elasticsearch’s classic TF/IDF [5] and the BM25 similarity algorithm (cf. section 2.1). For the conversion of the document ranking into a journal ranking—or, more general, collection ranking—four different voting schemes are used (cf. section 2.2).

2.1 Document ranking

**TF/IDF.** The score\((d, q)\) of a document \(d\) given a query \(q\) which consists of terms \(t\) is computed as follows \(score(d, q) = \sum_{t \in q} (tf(t \text{ in } d) \cdot idf(t) \cdot norm(d))\). The term frequency \(tf\) of term \(t\) in document \(d\) is computed as \(tf(t \text{ in } d) = \sqrt{\text{frequency}}\). The inverse document frequency \(idf\) for term \(t\) is computed as \(idf(t) = 1 + \log \left(\frac{\text{numDocs}}{\text{docFreq}(t) + 1}\right)\), where \(\text{numDocs}\) is the number of all documents in the collection and \(\text{docFreq}(t)\) is the number of documents containing term \(t\).
The normalization factor $\text{norm}(d) = \frac{1}{\sqrt{\text{numTerms}}}$ for a matching document $d$ causes higher weights for short documents (documents with a lower number of terms $\text{numTerms}$) in the score computation.

The score formula for Lucene’s classic similarity TF-IDF in addition contains other weighting factors (for normalization and coordination) which are not considered or not relevant in our experiments and not involved in the score-computation.

**BM25.** For BM25, the $\text{score}(d, q)$ of a document $d$ given a query $q$ which consists of terms $t$ is computed as follows:

$$\text{score}(d, q) = \sum_{t \in q} \left( \frac{\text{idf}(t) \cdot tf(t \text{ ind } d) \cdot (k + 1)}{tf(t \text{ in } d) + k \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \right)$$

The term frequency $tf$ describes the number of occurrences of term $t$ in document $d$, $|D|$ represents the document length, and $\text{avgdl}$ is computed as the average document length over all documents in the collection.

Here the inverse document frequency $\text{idf}$ for term $t$ is computed as $\text{idf}(t) = \log \left(1 + \frac{\text{numDocs} - \text{docFreq}(t) + 0.5}{\text{docFreq}(t) + 0.5}\right)$, with $\text{numDocs}$ and $\text{docFreq}(t)$ defined as before.

In our experiments, we use BM25 with standard values for $k$ (1.2) and $b$ (0.75).

### 2.2 Collection ranking

Based on the article ranking as search result, four approaches introduced in expert search are adopted to derive a journal ranking—respectively, a collection ranking. In general, the voting model can be based on different inputs: the number of items in the search result associated with a collection, the ranks of the items associated with a collection, and the score values calculated for the items associated with a collection [1].

Let $R(q)$ be the set of articles retrieved for the query $q$ and $\text{score}(j, q)$ the computed score for journal $j$ and query $q$, we apply four different voting models:

**Votes.** This metric takes the number of found articles for every journal as the score:

$$\text{score}_{\text{Votes}}(j, q) = |\{\text{article} \in R(q) \cap \text{Journal}(j)\}|$$

**CombSUM.** For every journal, this aggregation sums up the scores of the articles:

$$\text{score}_{\text{CombSUM}}(j, q) = \sum_{\text{article} \in R(q) \cap \text{Journal}(j)} \text{score}(\text{article}, q)$$

**CombANZ.** This aggregation is based on the previous CombSUM method normalized by the number of found articles for every journal (i.e. divided by the Votes-value):

$$\text{score}_{\text{CombANZ}}(j, q) = \frac{\text{score}_{\text{CombSUM}}(j, q)}{\text{score}_{\text{Votes}}(j, q)}$$
**CombMAX.** This metric takes the first result stemming from $j$, respectively, the article with the highest ranking, as voting candidate with its score:

$$\text{score}_{\text{CombMAX}}(j, q) = \max(\{\text{score}(\text{article}, q) : \text{article} \in R(q) \cap \text{Journal}(j)\})$$

3 **Experiment**

For our experiments, we take data from the dblp computer science bibliography (Digital Bibliography & Library Project), an online reference for bibliographic information on major computer science publications [4]. Dblp offers bibliographic metadata and links to the electronic editions of publications and consists of nearly 3,800,000 publications. The data is published by the University of Trier which exposes a dump for download and further research. In this work, we restrict the experiments using only the roughly 1,500,000 articles published in the 1,657 journals ignoring e.g. conference articles.

In the first step, we take 10,000 randomly chosen articles and remove them from the collection. All removed items serve as requests for the evaluation series with their title. The length distribution of all titles used as keyword queries is as follows: The minimum is 1 word. 25% of the queries/titles comprise 7 or fewer terms. The median is 9 and the 3rd quartile is 12. The longest title among the 10,000 chosen titles consist of 37 words.

For every item of the 10,000 articles, all combinations of the two similarity algorithms and the four voting models described in section 2 are combined and applied. Hence, every search request yields eight sets of ranked journals.

For each article and for each set of ranked journals the rank of the article’s corresponding journal is determined and saved. Ideally, an article’s corresponding journal should appear at rank 1 or at least among the top ranks of all ranked journals.

Our setup contains 1,657 different journals and nearly 1,500,000 articles. Figure 1 shows the distribution of articles over journals. Nearly 60% of all journals (965 items) contain 500 articles at most. As the maximum, the collection contains one journal in which more than 18,000 articles are published.

4 **Results**

For all 10,000 article-requests, we determine the rank of the article’s corresponding journal. Table 1 shows the effectiveness results for each combination of document-similarity model and voting technique. We state the 1st quartile (value $x$ meaning that for 25% of the queries the respective journal has been among the top $x$ results), the median, the 3rd quartile, the share of queries where the expected journal is in top 10, and the mean reciprocal rank (MRR).

Aggregations calculated by the Votes algorithm for BM25 and TF/IDF have the worst—and the same—results. For only 25% of the queries the journal associated with the requested article appears among the top 30 results. CombANZ and CombSUM follow, they deliver at least 25% of the corresponding journals within the top 20 results.
Fig. 1. All 1,657 journals categorized by the number of published articles

Table 1. Effectiveness measures for the rankings of the articles’ corresponding journals

<table>
<thead>
<tr>
<th>voting technique (combination)</th>
<th>BM25</th>
<th>TF/IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>voting technique (combination)</td>
<td>CombANZ</td>
<td>CombMAX</td>
</tr>
<tr>
<td>1st quartile</td>
<td>19</td>
<td>3</td>
</tr>
<tr>
<td>median</td>
<td>70</td>
<td>12</td>
</tr>
<tr>
<td>3rd quartile</td>
<td>242</td>
<td>59</td>
</tr>
<tr>
<td>share with journal in top 10</td>
<td>15.7%</td>
<td>48.5%</td>
</tr>
<tr>
<td>MRR</td>
<td>0.06</td>
<td>0.27</td>
</tr>
</tbody>
</table>

The best results for getting the associated journal are achieved by applying Comb-MAX which returns the searched journal among the top 12, respectively top 14, journals in 50% of all cases.

The presented results suggest that the underlying similarity algorithms (retrieval model) do not fundamentally change the ranking behavior of the imposed collection ranking methods. A much stronger influence can be observed for the combination schemes. Hence, these schemes seem to be worth more in-depth consideration.
5 Conclusion and Future Work

When comparing all ranking methods, it turns out that CombMAX yields the best results regarding the journal ranking. Here, no noise based on lower ranked results—respectively, articles—that contribute as voters is generated. In certain cases, these lower ranked results considered in the aggregation distort the ranking. When applying CombANZ and CombSUM, this effect is shown. Both methods yield worse results, especially regarding the rankings above the median and first quartile.

The approach using the Votes algorithm is not well suited in this form: not even 50% of the rankings see the expected journal under the top 100. This is because the approach does not differentiate between results on higher and on lower ranks.

CombMAX as the best performing approach is also in accordance with the experimental results presented in [1], when no profile length normalization was applied. Nevertheless, in our perception the results achieved by the simple CombMAX approach are surprising since the approach corresponds to a nearest neighbor based classification or a single link approach.

For further research, we plan to modify the applied aggregation techniques: according to current results, a CombSUM technique which considers only the upper ranking articles might deliver more accurate results. Considering only the top article result for every journal would end up in the applied CombMAX method.

Taking a closer look at the results it is a bit astonishing that CombSUM is much closer to Votes than to CombMAX. One could hypothesize that the decline of the score values might be well suited for ranking single documents but not for summing them up to yield a combined score for a journal. Problems with independence assumptions as well as ranking equivalent transformations in the formulas might be potential reasons in this respect. We plan to consider this further and to investigate combination schemes reflecting these observations. The sketched observations might also be the reason why using the reciprocal ranks from documents—respectively, articles—in combination schemes performed surprisingly well in [6].

6 References

How Do Search Engines Work? A Massive Open Online Course with 4000 Participants

Ralf Krestel and Julian Risch

Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Str. 2–3, 14482 Potsdam, Germany
firstname.lastname@hpi.de

Abstract. Massive Open Online Courses (MOOCs) have introduced a new form of education. With thousands of participants per course, lecturers are confronted with new challenges in the teaching process. In this paper, we describe how we conducted an introductory information retrieval course for participants from all ages and educational backgrounds. We analyze different course phases and compare our experiences with regular on-site information retrieval courses at university.

Keywords: Massive open online course, MOOC, search engines, IR teaching

1 Information Retrieval for the Masses

Massive open online courses (MOOCs) are established as a way of flexible learning with millions of participants worldwide. While the gold rush five years ago has lead to proclamations that MOOCs are the future of education [29], today, with a more mature field and years of experience with MOOC platforms, the goal and vision of MOOCs has shifted [35]. Instead of offering a complete curriculum, individual MOOCs are focusing on introductory lectures or specific technologies. Especially computer science topics are offered by various commercial platforms and universities. These courses are typically targeted towards (computer science) students and people working in the IT industry. These groups are used to online tutorials, e-learning environments, and flexible learning schedules. Also, they use MOOCs as an alternative to traditional courses at their university or off-the-job training to which most of them could get access to. But, the penetration of everyday life with Information, especially by the World Wide Web, necessitates a confident handling of new technologies not only by expert users. Search engines and information retrieval concepts in general are heavily used by people of all ages and educational backgrounds. While some high schools promote digital literacy, most people, especially older citizens, have never learned how the underlying technology they are using everyday works. This gap between high school graduates and IT experts can be filled by MOOCs as an alternative to adult education centers. Also, in countries where education is very expensive or not accessible or available, MOOCs provide a means of inexpensive teaching [8,13].

Information retrieval is not the only computer science topic relevant to a broader audience, but with search engines there exists a point of interaction that
everybody knows and uses daily. Therefore are information retrieval concepts and understanding how web search engines work important components for digital literacy in today’s societies. MOOCs on these computer science topics can be a vehicle to teach and inform all living in modern societies.

In contrast to traditional classroom teaching, a set of specific challenges needs to be addressed when teaching MOOCs. The most important one is the heterogeneity of the participants. Various age groups, cultural and educational backgrounds, and also motivation of the individuals differ largely. On the other hand, the technology of MOOC platforms allows usually to adjust the speed of learning individually and thus the trade-off between overburdening beginners and not challenging more experienced participants is delegated back to the user. In practice, this means recorded video lectures can be skipped or played back at a higher speed if the user is already familiar with the presented information.

The presentation of the lectures as short videos also has an effect on the preparation of the lecture. Maybe owing to the fast-paced modern times, people’s attention spans are getting shorter. The success of MOOCs relies — at least partly — on the very short lecture units compared to traditional university lectures of around 90 minutes. This condensed form of presentation in small chunks necessitates more precise wording from the teacher. Each phrase needs to be thoroughly planned to prevent misunderstandings which cannot be easily cleared up in contrast to classroom teaching. To handle open questions or deal with misconceptions, MOOC platforms offer space to discuss topics, ask questions, and in general interact with fellow participants and the teaching staff through a forum. This allows for deeper discussions or excursus to advanced topics, as well as to address specific aspects.

2 Search Engine MOOC on openHPI

We offer a Master’s lecture on “Information Retrieval and Web Search” where students not only learn information retrieval concepts but also design and implement their own search engines in small teams. In addition, we offer an introduction to information retrieval and web search for high school students in a three sessions course\(^1\). To fill the gap between teens and students and give interested persons outside the education system the opportunity to learn about IR, we offer a MOOC on the topic on the openHPI platform.

2.1 openHPI

The openHPI MOOC platform\(^2\) offers the interested public German-language and English-language online courses with a diverse computer science focus. While some courses target a broad audience and introduce fundamentals of computer science, other courses go into more detail of specialized, advanced topics.

\(^1\) https://hpi.de/open-campus/schuelerakademie/schuelerkolleg.html
\(^2\) https://open.hpi.de/
Course topics range from programming languages, mathematics, and IT law, to hardware-related topics, such as in-memory databases and mainframes.

The online learning platform is composed of videos, forums, quizzes and interactive programming environments. Figure 1 shows a screenshot of the website's video player, which is the starting point of all interactions on the platform. HPI actively conducts research on MOOCs and incorporates research results. For example, research suggests that videos should be segmented into short chunks (less than 6 minutes) and that the instructor’s head should be shown together with the presentation slides to increase student engagement [12].

![Screenshot of the course platform website’s video player.](image)

After watching the video of a learning unit, students can ask questions or discuss the unit’s content in the forum. Furthermore, they can evaluate their learning achievement with quizzes. These quizzes are composed of multiple choice questions and multiple answer questions. In the former, students need to choose the single right answer out of four possible answers, whereas in the latter, students need to choose multiple right answers out of four.

At the end of each course, there is a final exam. A record of achievement is issued to those students who have earned more than 50% of the maximum number of points for the sum of all graded assignments. A confirmation of participation is issued to those who have completed at least 50% of the course material.

### 2.2 Search Engine MOOC

The standard MOOC courses run for six weeks and cover content equivalent of a two hours per week semester course. Since we did not want to transfer our
The two-week course covers a broad topic range in 17 videos.

<table>
<thead>
<tr>
<th>1st Week</th>
<th>2nd Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1. Search queries and user interaction</td>
</tr>
<tr>
<td>2. History of information retrieval</td>
<td>2. Interaction: query processing</td>
</tr>
<tr>
<td>3. Text processing</td>
<td>3. Interaction: query refinement</td>
</tr>
<tr>
<td>4. Index construction</td>
<td>4. Interaction: search engine result pages</td>
</tr>
<tr>
<td>5. Ranking: Boolean retrieval</td>
<td>5. Web search</td>
</tr>
<tr>
<td>7. Ranking: factors</td>
<td>7. Social networks</td>
</tr>
<tr>
<td>8. Evaluation in information retrieval</td>
<td>8. Link analysis</td>
</tr>
<tr>
<td>9. New tasks and applications</td>
<td></td>
</tr>
</tbody>
</table>

Master’s lecture but to develop an introduction course without any requirements for the participants, we chose the shorter two week format. Our course “How Do Search Engines Work?” started in May 2017 with over 4000 registered participants. In 17 short videos of about twelve minutes each, we teach simple concepts of information retrieval and web search. While our videos are considerably longer (average length of 12 min) than the recommended six minutes [12], they are shorter than the videos of other openHPI courses, which was positively noted by many participants.

The content covered is inspired by university Master courses on information retrieval, but stays on a very high level without introducing algorithmic details. Students learn, for example, how a search engine is built, which process is started when searching for something, and according to what criteria the results are listed. Table 1 lists the course topics per week. Because of the student’s different levels of familiarity with mathematical proofs and probability theory, the online course does not cover the details of iterative computation of PageRank, index compression, or probabilistic information retrieval.

After each video, students can take a short, optional quiz as a selftest. This quiz can be repeated as often as wanted. Figure 2 exemplifies a selftest quiz after a learning unit about link analysis and the PageRank algorithm. To answer this question, students need to rank the nodes in this graph by their PageRank value. For this, it is not necessary to compute exact values but it is sufficient to have understood the general mechanism behind PageRank.

At the end of the two-week course, there is an exam with 16 questions. The exam time is limited to one hour and once started, cannot be paused or restarted. Neither the selftest quizzes nor watching the videos are needed as qualification to take the exam. However, for preparation, students can recap the course content with randomly sampled sets of selftest quizzes. An advantage of this form of learning is the instant feedback. Immediately after submitting the quiz, students get their evaluation together with recommendations of videos they may want to revise.
Fig. 2. Exemplary selftest question: Which of the node lists is ordered by PageRank with regard to the graph given on the left?

Table 2. In an optional survey, most students indicated to have enrolled for extended vocational training, whereas only a minority chose academic training as their reason for enrollment.

<table>
<thead>
<tr>
<th></th>
<th>Vocational Training</th>
<th>Spare Time/Interest</th>
<th>Academic Training</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Students</td>
<td>700</td>
<td>636</td>
<td>72</td>
<td>54</td>
</tr>
</tbody>
</table>

3 Course Participants

Compared to regular information retrieval courses at university, the audience of this introductory MOOC was very heterogeneous. Furthermore, there were no pre-requisites for this course. While students indicated different reasons for their enrollment, the majority took the course for extended vocational training as shown in Table 2. In total, 4458 students enrolled, of whom 1135 watched the last video of the course and of whom 698 took the exam.

Fig. 3. The number of students watching the video and taking the quiz per course day falls from over 2300 to less than 1000.
3.1 Statistics

Of 4500 enrolled students, 2500 participated in the course (watched at least one video) and 883 took the final exam. Of these 883 students, 19% were female and 81% were male. The data basis for the following statistics are only those students, who participated in the exam. Figure 4 shows the distributions of their age and highest degree. In Figure 5, we visualize the distribution of the number of points achieved in the final exam grouped per highest degree. Each group covers a wide range of achieved points and interestingly, the median number of exam points supports the assumption that degree and number of exam points correlate. In addition, we analyze the correlation of the number of exam points and the number of visited videos and quizzes. Under the assumption of a linear correlation and according to a linear regression, we find a significant correlation of these variables. A visited video or quiz correlates with an increase by 0.2 points in the final exam.

![Age vs Frequency](image1)

![Highest Degree vs Frequency](image2)

Fig. 4. Distributions of age and degree among 883 students who took the final exam

3.2 User Participation

A key advantage of MOOCs is the larger number of course participants that can discuss the learning units together. In 72 different threads, students asked questions about the course content or delved into a subject with discussions. While there were 94 answers to questions and additional 78 comments to questions and answers, only a third of these replies came from the teaching team. Two-third of the replies to questions came from the course participants themselves. Research has shown that a successful outcome is not correlated with teaching team interaction in the forum [33]. In total, there were 358 posts in the forum and 6479 views of these posts. This confirms findings by [2]: they reveal usage peaks on weekends (spare time available) and that over 90% of forum activity are passive views and only 10% are active posts. Discussed topics covered various types:
from technical questions, to ethical issues, and personal anecdotes. One of the most discussed topics was the bag of words model, where participants posted different examples for texts and their respective bag of words. They discussed how lemmatization and stemming can change the bag of words and consequently change also the results of search queries. But also non-technical discussions arose, e.g. about the German plural form of the word “index”.

A different discussion focused on alphabetical subject catalogs and the difference to tags and keywords. An older participant asked: “Is it really true that most people don’t know alphabetical subject catalogs?” While some participants were indeed not aware of library catalogs (“I know these [alphabetical subject catalogs] only from databases and electronic systems.”), others remembered public libraries with index cards (“As high school student and later on as university student, catalogs were quite common for me. After watching this video, I am feeling old.” or “When I was a student (70s/80s), the introductory week for new students included getting to know the library and searching for books.”). Luckily, experts in the field joined the discussion: “I am librarian and of course tagging is still used. However, most people use keyword search in electronic databases and do not notice tagging.”

4 Research Challenges on MOOC Platforms

In addition to their educational purpose for course participants, MOOC platforms provide various research opportunities. The vast amount of data gives insights into the learning process. Under the umbrella of e-learning, learning analytics, and educational data mining a research community has formed looking at various aspects of MOOCs and trying to improve the learning experience.
Analyzing MOOCs: An initial study of the data generated by MIT’s first MOOC reports on characteristics of the students and their use of course resources [2]. Insights into a recommender systems MOOC [21] and a German database MOOC [28] confirm the findings. Also, enrollment numbers and success rates can be analyzed in detail [24].

Infrastructure/Platform Guo et al. study how MOOC video production affects student engagement [12]. Also other elements were the focus of research work, e.g. the forum and whether gamification is beneficial [5].

Tools for Teachers Designing and evaluating tools to support teachers is important. Helpful in this context is also to know more about your (potential) participants. Chen et al. match course participants with their profiles in several online social networks [4]. Thereby, they are able to compare course topics with job titles of participants. As a result, teachers learn more about students and their educational needs. Similarly, the effectiveness of certain tools needs to be evaluated, e.g. whether instructor involvement in forums has any impact on student outcomes [33]. Also prediction of student success or assessment of progress needs to be monitored, e.g. by analyzing natural language texts [6].

Student Behavior Modeling Understanding students’ behavior is necessary to ensure a good learning experience. This starts with recommending or predicting course enrollment [27]. Once students are enrolled, keeping them motivated and helping them learn is most important. Drop-out rates are overall still high and therefore predicting drop-outs can help to intervene at the right point to keep students interested [18]. During courses, drop-out prediction can also help understanding at which point in the course users discontinue [26]. Analyzing discussion forums with respect to sentiment [34] revealed that the daily drop-out rate and sentiment expressed in forums correlate significantly. These students-at-risk should be identified and supported [14]. In addition to forums, clickstream data can be used to detect changes in student behavior [30].

Collaborative Learning The social component of MOOCs plays an important part for successful learning. Peer grading is very important for large courses and the quality of peer grading can be increased by motivation [22]. The authors show that students can be motivated to put more effort into peer grading if they are confronted with the effort that other graders put into it and if they are asked to evaluate other graders’ efforts. Recently, the social connections between course participants and their influence on peer assessment have been studied [3]. Social comparison can be used to increase peer pressure to complete a course and reduce drop-out rates [7]. In forums, answers to questions can be ranked based on helpfulness automatically to reduce the load for other users to find the correct or best answer [17].

Personalized Learning The future of MOOCs is probably the personalization of the learning process. Individual participants have different learning strategies, speed, and learning preferences. This can be a huge advantage over traditional classroom teaching. First approaches for adaptive learning are already tested [32]. Further, the consideration of mood and emotions within
the learning process can help to reduce drop-out rates. Affective learning is one approach to this end [31].

5 Teaching Information Retrieval

Before the age of the MOOCs, e-learning environments existed that were also used for teaching information retrieval. Henrich and Morgenroth [15] report on their experience with using different e-learning scenarios in the context of IR. Similar to our experience, the forum was an important mean of communication among students and between students and faculty. We also offer the possibility for students to perform self-tests in the form of multiple choice questions after each lesson. With respect to the learning material, we provide the slides as PDF-documents along with the videos. Regarding the ordering of topics in the syllabus, we first introduced classical information retrieval concepts and in a second part elaborated on the specific Web IR challenges, as recommended by Mizzaro [25]. Henrich and Stieber [16] further analyzed IR e-learning courses along two dimensions: degree of interaction (e-learning vs. blended learning) and main media type (text-based vs. recording-based). The results indicate that all combinations can work out and other factors are more important for success, such as a clear teaching concept tailored towards the specific target audience, or active participation of students and lecturers in forums. Other papers [1, 10, 23] discuss IR curriculums and possible accompanying practical exercises for full term IR courses. Kauchak [20] reports about his experience with a course-long project consisting of the development of a search engine. While this is something that we also do for our regular IR courses, this is clearly not feasible for a short-term, introductory MOOC course. Also alternative teaching methods (inquiry-based learning) were applied to full term IR courses [19], supporting the learning-by-doing paradigm. Many related work appeared in two workshops 2007 and 2008 with the title "teaching and learning in information retrieval". In addition, in 2011 a book [9] with the same title was published covering many aspects of IR teaching in various chapters.

6 Lessons Learned

Some students missed course content about more advanced use of search engines and more advanced search operators. Beyond the topics of our course, students wanted to learn about search engine optimization or online reputation management for search engine results. Several students asked for practice-oriented hands-on exercises. This feedback was retrieved also for other courses on the openHPI platform and openHPI is now experimenting with concrete exercises in server-based training environments [11]. For example, the course “Data Management with SQL” incorporates practical tasks where students create SQL queries for MySQL [28].

Regarding mathematical backgrounds of the presented models, several students praised comprehensive explanations without too much detail for this short
two-week course. They aimed at a general understanding of search engines and
preferred concrete examples. The exemplary indexing or ranking of documents
and the expansion of queries was perceived more informative than theoretical
evaluation measures, such as MAP or NDCG. Although the two course weeks
have ended, the course material stays available. For this reason, we expect the
enrollment number to increase further.

There seems a demand for short introductory courses in IT especially from
older participants. To allow flexible learning, a single video shouldn’t exceed 10
minutes in length and the video length variance should be small for individual
planning reasons. Also longer MOOC courses (six weeks) on IR with more details
are possible to increase the level of detail and add some more practical examples
and homework tasks.

References

Uncovering Business Relationships: Context-sensitive Relationship Extraction for Difficult Relationship Types

Zhe Zuo, Michael Loster, Ralf Krestel, Felix Naumann

Hasso Plattner Institute
Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany
{firstname.lastname}@hpi.de

Abstract. This paper establishes a semi-supervised strategy for extracting various types of complex business relationships from textual data by using only a few manually provided company seed pairs that exemplify the target relationship. Additionally, we offer a solution for determining the direction of asymmetric relationships, such as “ownership_of”. We improve the reliability of the extraction process by using a holistic pattern identification method that classifies the generated extraction patterns. Our experiments show that we can accurately and reliably extract new entity pairs occurring in the target relationship by using as few as five labeled seed pairs.

1 Business Networks

Extracting structured data from text, and thus harnessing the valuable information on the web and hidden in the vast amounts of other textual data, is a well-known and well-studied research area. As the text corpora and the kind of information to be extracted from them can vary greatly, many research works have focused on specific types of information, on specific corpora, on specific application domains, on specific languages, or any combination of the above. In this paper, we regard the problem of extracting relationships of several specific types among companies from news articles.

Many tasks, such as building business networks, predicting risks, or valuating companies, can significantly benefit from accurately extracting relationships between companies. Imagine a scenario in which Dell wants to acquire EMC. Dell plans to finance the deal by taking out a loan. The chosen bank has to decide whether to award the loan based on the careful assessment of the risk associated with this transaction. With the explosive growth of the textual data on the web, it becomes possible to discover not only the information of Dell and EMC but also the dependencies by extracting business relationships and building up a company network. In the same example, by analyzing the network structure of both companies, the bank might reach the conclusion that the risk of granting a loan is too high, because many of EMC’s subsidiaries, as given by the relationship network, are struggling. With this knowledge the bank might award a smaller or no loan at all or propose a higher interest rate.

To build up a business network between companies, it is critical to reliably extract business relationships. Companies often connect to each other via the activities in which they participate. Business relationships represent a subset of these activities; examples include ownership_of, partnership_with, supplier_of, and so on. Only very few of them
can be found in structured knowledge base like Freebase [4] or semi-structured data like Wikipedia infoboxes – a substantial amount of relationships is hidden in unstructured data sources. Aggravating this situation, both Freebase and infoboxes contain only the major subsidiaries of some companies (i.e., ownership of relationship). Other relationships, such as partnership with or supplier of, are not covered.

Given a corpus of unstructured textual data, we aim to (1) discover whether two co-occurring companies participate in a business relationship, (2) identify the type of the relationship, and (3) in the case of an asymmetric business relationship, determine its direction.

The task of business relationship extraction is challenging due to the complex nature of the relationships between companies. First, multiple types of relationships can exist between two companies. Samsung as one of the biggest competitors of Apple is also the supplier of displays for Apple’s products. Moreover, as an example of resolving the direction of asymmetric relationships, such as the ownership of relationship, consider that Walt Disney owns ABC Studios but not the other way around. Being able to successfully derive the direction of the relationships is of vital importance for many subsequent tasks.

The Snowball system addresses the general problem of relationship extraction [1], and our work is based in parts on its general idea. It takes a small set of entity pairs as a seed set and generates candidate patterns that are based on the context of these pairs. Subsequently, the most prominent patterns are selected according to a scoring function and used to extract new entity pairs that participate in the target relationship. In the end, the newly selected pairs are added to the seed set and the process repeats to generate more patterns. However, Snowball functions only correctly if there is a one-to-many relationship between the participating entities, e.g., in the headquarter of (Microsoft, Redmond) relationship, Microsoft has exactly one headquarter. Business relationships do not adhere to this characteristic, which is the reason Snowball is unable to solve the problem at hand.

We extend the Snowball idea by introducing a key-phrase extraction strategy, which allows us to remove irrelevant parts of the context surrounding the company pairs. To determine the direction of asymmetric relationships, we propose a process that leverages information contained in the seed set. Since Snowball cannot deal with many-to-many business relationships, we propose a generalization of their tuple- and pattern-evaluation strategy by specifying a new selection method to select patterns and new seeds. We further define a holistic pattern identification strategy, which enables us to extract multiple relationship types simultaneously.

In summary, we propose a system to perform (directed) relationship extraction (RE) between companies from textual data. Addressing this problem, we present a novel, semi-supervised relationship extraction method, which requires only a minimum amount of manually specified company pairs to efficiently extract new ones that belong to the same target relationship. Additionally, we provide a straightforward solution to reliably identify the direction of asymmetric relationships. We show that our approach is superior to more advanced distant learning approaches for the particularly difficult case of many-to-many relationships.
2 Background and Related Work

The most related work is the Snowball system [1], which we have already introduced in Section 1. Despite the fact that there is a large body of work that focuses on the topic of relationship extraction, the subject of extracting business relationships between companies from unstructured data has not been sufficiently addressed by research.

One way to approach the general relationship extraction problem is to use supervised learning techniques by classifying whether two entities participate in a specific relationship. Kambhatla [6] employed Maximum Entropy models to solve the relationship extraction task. Zhou et al. [13] also applied a feature-based relationship extraction strategy that uses Support Vector Machines (SVM) [8]. Further, kernel methods with string-kernels have successfully been applied to deal with the RE problem [11]. The major drawback of these techniques is that a large amount of labeled data is required for training. As a representative example, Kambhatla [6] uses a training set that contains around 9,752 instances of relationships to generate their results. Moreover, relabeling and retraining of the model becomes necessary, as soon as either the underlying characteristics of the data sources or the target relationship change substantially.

Mintz et al. [7] introduced a distant supervision approach, which avoids the expensive labeling process. The idea is to automatically label the training data according to the relationships included in knowledge bases, i.e., Freebase [4]. One of the limitations is that it is highly rely on the given knowledge base, only the types of relationships that are included can be extracted, while most of the business relationships are not covered at all, such as partnership_with, competitor_of and supplier_of.

Another way to address the problem was presented by Banko et al. [2]. They introduced an unsupervised approach called TextRunner to extract all possible relationships in a given corpus without requiring any labeled data. This task is known as the open information extraction (Open IE) task. Wu and Weld [10] proposed the WOE system, which enhances TextRunner by including additional information from Wikipedia infoboxes to construct a training dataset. Although the Open IE approaches can automatically extract all possible relationships from a given corpus, their results cannot directly be used in further applications. They can neither disambiguate mentions nor provide semantic information about the extracted relationships automatically.

We avoid labeling large amounts of training data and predefining a specific type of relationship by using a few examples of a target relationship for bootstrapping. This idea was first introduced by Brin [5] in the context of the DIPRE system, which focused on extracting relationships between authors and their corresponding book titles. Some other approaches were developed based on this bootstrapping strategy, e.g., Snowball [1] and StatSnowball [14].

We focus on reliably extracting business relationships between companies. By applying the semi-supervised algorithm, we can extract more complicated many-to-many relationships from large amounts of unlabeled data without requiring the expensive initial labeled data. A user only has to supply a very small number (3–5) of seeds to achieve good results, which makes our approach flexible to be applied to variant target relationships or data sources by simply provide another small seed set. Furthermore, our approach is able to determine the direction of asymmetric relationships. This enables us to directly use the generated results in subsequent applications.
3 Overview of our Approach

Figure 1 gives a high-level overview of our relationship extraction approach: Given some textual data and a seed set of multiple company pairs that occur as members of a particular relationship, our system outputs new company pairs participating in the same relationship type. As a preprocessing step we simplified the algorithm introduced by Zuo et al. [15] to recognize and link the mentions of companies to their corresponding Wikipedia pages.

Given these disambiguated company mentions we generate patterns from their contexts. To this end, we follow the intuition that if a company pair from the seed set co-occurs in the same sentence, it is likely that the context characterizes the relationship specified by the seed (see Section 4.1). Therefore, the sentences that contain two or more distinct companies are selected as the input for the relationship extraction phase. An example tagged sentence is “...[Verizon Communications|Verizon]’s acquisition of [[MCI Inc.|MCI]]”, where the original mentions “Verizon” and “MCI” are separately linked to Verizon Communications and MCI Inc. From the contexts surrounding company pairs, we generate possible candidate extraction patterns that are likely to represent the target relationship (see Section 4.1). Suppose we are interested in the ownership relationship and the company pair (Verizon Communication, MCI Inc.) is contained in the seed set, then a candidate pattern $\text{pattern} = \langle \text{COMP}_1, \text{COMP}_2, \text{acquisition of, } \rightarrow \rangle$ can be generated. The last element in $\text{pattern}$ describes the direction of the ownership relationship (see Section 4.4). After generating a list of candidate patterns, we select the most promising ones according to the measurements to be introduced in Section 4.2.

We then use the selected patterns to discover new company pairs from the input. If the previous pattern $\text{pattern}$ is selected, we can extract a new company pair (The Walt Disney Company, Pixar) from a sentence like “...after Disney’s acquisition of Pixar Animation Studios”. Afterwards, we select the most prominent newly extracted pairs to extend the seed set (see Section 4.3). We then iterate the procedure to extract new patterns using the extended seed set until no more new company pairs can be selected as seeds or the iteration number reaches a predefined limit. The company pairs that are extracted based on the current set of patterns are considered to participate in the same type of relationship as the target one. Our evaluation shows that this is indeed almost always the case regardless of the initial choice of seed pairs.
4 Extraction of Business Relationships

This section introduces our semi-supervised relationship extraction strategy, which iteratively extracts new company pairs that participate in a given target relationship.

4.1 Pattern generation

Generating the extraction patterns represents a crucial step in our approach. The context surrounding a company pair represents the main source to identify relationships occurring in textual data. To capture the key information that represents the relationship between two companies we extract the most determining phrases from the context as a key-phrase. This key-phrase is then used to generate a pattern.

Candidate pattern An extracted pattern includes two company variables COMP1 and COMP2, the key-phrase extracted from the context in between those companies, and a direction. We explain each of these parts in the following. From an example sentence “...YouTube, the video-sharing Web site owned by Google...” we can generate the pattern ⟨COMP1, COMP2, owned by, ←⟩. By applying this pattern to this sentence we obtain the following instantiation of the pattern ⟨YouTube, Google, owned by, ←⟩, indicating that Google owns YouTube.

Key-phrase extraction The quality of a pattern depends on the key-phrase it contains. A good pattern should satisfy two criteria: characterize a single type of relationship (which in turn improves the precision of the extraction result) and be as general as possible (to extract many new company pairs). For this reason, it is beneficial to generalize the context and don’t keep idiosyncratic key-phrases. The key-phrase should be as compact as possible, while maintaining the information in the context. Extracting patterns for business relationships in the news is particularly challenging since journalists are used to introduce the same type of business relationships using different writing styles spanning a relatively large context. This can be shown using the excerpt “...News Corporation, which owns a minority interest in DirecTV”. In this sentence, we can easily figure out that News Corporation is one of the owners of DirecTV by finding the verb “owns” in the intermediary context. If we now use the entire context (i.e., “...which owns a minority interest in’’) between the two companies as a pattern to extract additional company pairs, we would find only very few since the pattern is not general enough. The problem can be solved by extracting the key-phrase “owns” that defines the ownership relationship. Thus we can conceptually simplify the original sentence to “New Corporation owns DirecTV”. To this end, we developed a key-phrase extraction strategy to automatically extract the most determining phrases from the intermediary context. Intuitively, relationships in sentences are often conveyed by verbs or nouns. In [3] most of the binary relationships are indicated by four types of phrases, which cover over 86% of the cases. These key-phrase types are “Verb”, “Noun+Prep.”, “Verb+Prep.”, and “Infinitive” located in between two entities in English text. To extract key-phrases from contexts, we apply the Stanford Part-Of-Speech(POS) Tagger 1.

1 http://nlp.stanford.edu/software/
Based on the POS tags, we keep the phrases that match any of the four types above. We abandon the context if the containing verb is “to be”, because it usually does not indicate any business relationship, or if the context contains multiple key-phrases.

### 4.2 Pattern selection

In each iteration, we generate candidate patterns based on the (extended) seed set. However, patterns that do not represent the target relationship might also end up in the candidate list. Therefore it is important to keep only the most representative patterns while filtering out unfavorable patterns. In the following, we introduce two strategies to select the best patterns.

**Hit score** Building on the intuition that patterns that frequently match company pairs in the seed set are likely to be representative ones, we introduce a *Hit score* for each pattern as follows

\[
\text{Hit}(\text{pattern}|\text{Pair}_\text{seed}, S) = \sum_{\text{pair}_i \in \text{Pair}_\text{seed}} \sum_{s_j \in S} \text{match}(\text{pair}_i, p, s_j)
\]  

Thus *Hit* is defined as the summation of how frequently a pattern matches a company \(\text{pair}_i \in \text{Pair}_\text{seed}\) in the set of input sentences \(S\). A pattern with a high *Hit* score denotes that the corresponding key-phrase is more likely to represent the target relationship. Given a list of candidate patterns that are sorted in descending order by their respective *Hit* score, we select the top-\(k\) ranked patterns to extend the set of the current extraction patterns.

**Coverage score** A good pattern should frequently be used in the context between different company pairs to describe a particular relationship. If the pattern can be extracted by using only one of the seed pairs, it is either too specific or it describes some other type of relationship between the corresponding companies. We introduce a Coverage (*Cov*) score, which represents the percentage of company pairs from the seed set that are able to generate this pattern.

\[
\text{Cov}(\text{pattern}|\text{Pair}_\text{seed}, S) = \frac{\sum_{\text{pair}_i \in \text{Pair}_\text{seed}} \sum_{s_j \in S} [\text{match}(\text{pair}_i, p, s_j) > 0]}{|\text{Pair}_\text{seed}|}
\]

The *Cov* score of a pattern equals 1.0 when all seed pairs match the pattern at least once. All patterns that have a *Cov* score greater than a threshold \(\tau\) are selected.

### 4.3 New seeds selection

We introduce a similar strategy for selecting newly extracted company pairs to extend the seed set. Using the selected patterns, we compute the *Hit* score for each of the extracted company pairs. We select the top-\(k\) pairs by their *Hit* scores. We can also compute the *Cov* score of an extracted company pair, which is the percentage of selected patterns that match the company pair in the text. In a similar fashion to the pattern selection, we extend the seed set by selecting the company pairs that have a greater *Cov* score than the same given threshold \(\tau\) for pattern selection.
4.4 Direction of relationship

In Section 1 we introduced the challenge of determining the direction of asymmetric business relationships. Compared to the extraction of symmetric relationships, extracting asymmetric ones, such as supplier_of, ownership_of, and sued_by require not only the extraction of a new company pair occurring in the target relationship, but also the detection of its correct semantic direction.

Previous work, such as Snowball [1], naturally avoids this direction problem, since they focus on relationships that relate two objects of different entity types (i.e., organization, location). However, in our case, the entities are of the same type (i.e., company). Zhu et al. [14] present a similar challenge, e.g. an entity of type person $e_1$ is the husband of $e_2$. They solve this problem by manually adding new rules, such as IsHusband($e_1, e_2$) $\Rightarrow$ IsWife($e_2, e_1$), during their iterations.

We introduce an elegant strategy to automatically classify the direction of newly extracted relationships. The idea is to include the direction information already in the seed set. When the target relationship is asymmetric, the company pairs in the initial seed set must be specified by also providing the direction of the relationship. E.g., in the case of the ownership_of relationship, we specify a forward direction, denoting that the first company is the owner of the second.

Given this directed seed set, we can identify the direction of the generated patterns as follows: When two companies are mentioned in the same order as in the seed pair, the pattern is annotated with the same direction as the seed pair. Finally, the direction of a pattern is derived by assigning the direction that is more frequently marked. Table 1 in the evaluation section shows some examples of determined directions of patterns.

4.5 Multiple types of relationships

With our semi-supervised business relationship extraction approach, we can independently extract different relationship types by providing multiple initial seed sets each characterizing one type of relationship.

As mentioned in Section 1, different types of business relationships can exist between two companies at the same time. Therefore, the patterns generated from the seed set do not always represent the desired relationship type. Even worse, once a pattern that represents an undesired relationship type is selected, the following iterations can be negatively influenced in a way that they yield more and more irrelevant patterns, which leads to incorrect extraction results comparable to a topic drift in pseudo-relevance feedback methods. We can avoid this problem by assigning each pattern that is generated for multiple relationship types exclusively to one single type.

We followed the intuition that each pattern characterizes one kind of relationship and implemented a holistic pattern identification strategy by using the Cov score. In case the same pattern is generated for multiple relationship types, we exclusively assign the pattern to the type that yields the highest Cov score.

As a preliminary experiment to show the effect of this holistic strategy, we applied our approach to extract the ownership_of and partnership_with relationships at the same time. The selection of patterns and new seeds was made using the Hit score. By applying the holistic pattern identification strategy, most of the patterns, especially
the top-ranked ones, characterize the partnership with relationship. Without our holistic strategy, the top-ranked patterns (i.e., “stake in”, “deal with”, and “buy”) mainly represent the ownership of relationship. This problem was caused by a falsely selected pattern (i.e., “owned by”), which led to more and more patterns that characterize the ownership of relationship.

5 Experiments

In our evaluation we focus on the extraction of an asymmetric relationships (i.e., ownership) from articles of the New York Times corpus.

5.1 NYTimes corpus and seeds

The full New York Times corpus contains 1,855,658 news articles, spanning a period of 20 years from Jan. 1987 to Jun. 2007. We observed that about 74% of all company pairs within a sentence occurred in the “Technology” and “Business” categories. Thus, we reduced our corpus to articles with at least one of those two labels. Our final corpus (called NYTimes from now on) consists of 359,459 articles.

An initial seed set serves as the input for our approach and predefines the relationship type we would like to extract. We investigated two different seed sets to evaluate their influence on the results. To this end, we generated a list of distinct company pairs that co-occur in the NYTimes corpus and sorted it in descending order by co-occurrence frequency. We manually labeled the relationship type for the first 100 pairs and then randomly selected five company pairs (FreqSeed) that share the ownership of relationship from the top-100 list entries. Following this random selection strategy, we also generated a seed set called InfreqSeed from the top-1000 company pairs. Keep in mind that seed selection and our evaluation is based on a corpus dating from 1987 to 2007, resulting in relationships that might not hold today. FreqSeed contains company pairs, such as (AOL, Netscape), (Viacom, Viacom Media Networks), (Ford, Jaguar), (Time Warner, TBS), and (GE, NBC Sports), while InfreqSeed contains less frequently mentioned pairs, such as (Disney, ESPN), (IPC, Campbell Mithun), (GM, Saturn), (Chrysler, American Motors), and (Investcorp, Saks Fifth Avenue).

5.2 Experimental results

We first show which patterns were generated and then evaluate the quality of the actual business relationships we extracted.

Results of pattern generation Based on the two randomly generated seed sets we applied our approach to extract new company pairs that are also members of the ownership of relationship. Table 1 shows the key-phrases of the selected patterns that are automatically generated by using FreqSeed and InfreqSeed. In this experiment, we applied the Hit score in each iteration for selecting the top-10 candidate patterns and the respective company pairs. The first column shows the key-phrases of the selected patterns. By using either FreqSeed or InfreqSeed, the extraction process terminates after
Table 1. Key-phrases of selected patterns for extracting ownership_of relationship.

<table>
<thead>
<tr>
<th>Extracted Patterns (key-phrase)</th>
<th>Rank (Iteration)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FreqSeed</td>
<td>InfreqSeed</td>
<td>FreqSeed</td>
<td>InfreqSeed</td>
<td>Direction</td>
<td></td>
</tr>
<tr>
<td>unit of</td>
<td>1 1 1</td>
<td>4 1 1</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parent of</td>
<td>- 4 2</td>
<td>- 4 2</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>owned by</td>
<td>2 2 3</td>
<td>1 3 4</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>part of</td>
<td>4 3 4</td>
<td>2 2 3</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>division of</td>
<td>5 5 5</td>
<td>3 5 5</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>owns</td>
<td>3 6 6</td>
<td>7 6 6</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>company of</td>
<td>- 7 9</td>
<td>- 8 9</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acquisition of</td>
<td>7 7 8</td>
<td>6 7 7</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subsidiary of</td>
<td>- 8 9</td>
<td>- 8 9</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>owner of</td>
<td>- 10 10</td>
<td>- 10 10</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>including</td>
<td>9 9 11</td>
<td>- - -</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>include</td>
<td>8 10 12</td>
<td>- - -</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bought</td>
<td>10 11 13</td>
<td>9 11 12</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acquired</td>
<td>6 12 14</td>
<td>5 12 13</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>buy</td>
<td>- - -</td>
<td>10 10 11</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bought by</td>
<td>- - -</td>
<td>8 13 14</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Three iterations resulting in 14 selected patterns. These patterns are sorted in descending order by their Hit score. We also include the ranks of the patterns per iteration to show the changes that occur from iteration to iteration.

Further, Table 1 shows that most of the automatically generated key-phrases are typical phrases frequently used to describe an ownership_of relationship. Already in the first iteration, our approach can generate representative patterns. Differences between the two sets of generated patterns can be observed mainly in the tail. Thus, our approach is not particularly sensitive towards the chosen seed set (we made similar observations for various other seed sets, both in terms of size and co-occurrence frequency).

The last column in Table 1 contains the extracted direction of the patterns as determined by the strategy introduced in Section 4.4. Only 2 out of the 16 directions are incorrect. Although the direction of these two patterns is classified incorrectly, most directions of the newly extracted company pairs are identified correctly as the statistics in Section 5.2 show. This is because the direction of newly extracted company pairs is determined by multiple patterns.

Quality of extraction results We applied our approach using different settings for both pattern and seed selection to verify the extraction result. We conducted experiments with the Hit and Cov scores strategies introduced in Section 4. To show the effect of our key-phrase extraction strategy, we also executed our algorithm without using this strategy. In other words, we employed the original context to generated patterns, which is similar to previous work, e.g., [1, 5]. As a baseline, we select the most frequently co-occurred company pairs to check how many of them are in an ownership_of relationship.

We had to manually check relationships between company pairs, because no gold standard with known business relationships is available. The design of our approach is
Table 2. Average precision and precision values for the top-50, top-100, and top-200 extracted ownership of relationships (including error analysis of the top-200 results)

<table>
<thead>
<tr>
<th>Strategy</th>
<th>P@50</th>
<th>P@100</th>
<th>P@200</th>
<th>Avg Prec</th>
<th>Error Type (Top 200)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>30.0%</td>
<td>36.0%</td>
<td>30.5%</td>
<td>28.3%</td>
<td>Rel. - - - - - -</td>
</tr>
<tr>
<td>Hit@10 w/o KP</td>
<td>18.0%</td>
<td>19.0%</td>
<td>20.0%</td>
<td>18.9%</td>
<td>139 4 4 13</td>
</tr>
<tr>
<td>Hit@5</td>
<td>94.0%</td>
<td>90.0%</td>
<td>88%</td>
<td>91.4%</td>
<td>14 0 8 2</td>
</tr>
<tr>
<td>Hit@10</td>
<td>94.0%</td>
<td>89.0%</td>
<td>87.5%</td>
<td>90.5%</td>
<td>9 4 7 5</td>
</tr>
<tr>
<td>Hit@15</td>
<td>94.0%</td>
<td>88.0%</td>
<td>85%</td>
<td>90.0%</td>
<td>9 8 7 6</td>
</tr>
<tr>
<td>Cov(τ = 0.7) w/o KP</td>
<td>20.0%</td>
<td>20.0%</td>
<td>23.0%</td>
<td>21.0%</td>
<td>135 3 3 13</td>
</tr>
<tr>
<td>Cov(τ = 0.6)</td>
<td>94.0%</td>
<td>87.0%</td>
<td>81.0%</td>
<td>88.7%</td>
<td>17 4 7 10</td>
</tr>
<tr>
<td>Cov(τ = 0.7)</td>
<td>94.0%</td>
<td>90.0%</td>
<td>90.0%</td>
<td>91.9%</td>
<td>8 1 6 5</td>
</tr>
<tr>
<td>Cov(τ = 0.8)</td>
<td>94.0%</td>
<td>88.0%</td>
<td>87.0%</td>
<td>90.6%</td>
<td>8 7 6 5</td>
</tr>
</tbody>
</table>

mainly concerned with achieving a high precision value because we aim to use it in the context of risk-analysis, which has only a small tolerance for incorrectly extracted information. Therefore, we mainly focus on evaluating the precision performance of our approach. We manually examined the top-200 most frequently extracted company pairs from each result set produced by our algorithm with different parameterizations.

Table 2 presents the evaluation results using the FreqSeed seed set to extract the ownership of relationships. As this table shows, by applying Cov (τ = 0.7) score, 90% of the top-200 extracted company pairs indeed participate in the ownership of relationship. The performance of our approach, excluding the key-phrase extraction strategy, also shows the significant effect of including it. In comparison to the baseline, our approach can produce much better results.

Apart from the precision measure, we also present a detailed error analysis based on the top-200 extracted company pairs: The first error type is that company pairs that do not participate in an ownership of relationship are extracted (Rel.). Another error case is that our approach extracted the correct company pair, but failed to identify the correct direction (Dir.). A third error case is caused by recognition or disambiguation errors made by the preprocessing steps (Pre.). An incorrect result can also be due to misinterpretation of the semantics (Sem.). E.g., one company finally canceled the plan of acquiring another one, such as the abandoned merger between EMI and Time Warner. Such events are covered by a series of New York Times articles, but our approach was unable to successfully capture the final cancellation of the deal. As the result shows, only around half of the incorrectly extracted relationships (i.e., Rel. and Dir.) are caused by our RE strategy.

Furthermore, according to the mechanism of our approach, when a relationship is mentioned in the given corpus more frequently, the probability that our approach can extract that relationship is higher. Thus, by including more documents the recall of our approach increases. We iteratively applied our approach (with the setting Cov (τ = 0.7)) to an NYTimes corpus of increasing size, starting from 10 years of data up to 21 years. In Figure 2, the red line denotes the total number of tagged sentences after our preprocessing step. The blue bars show the accumulated count of extracted company pairs. As the figure shows, by enlarging the size of the dataset more unique aimed relationships can be extracted.
We compared our approach with a state-of-the-art distant learning approach developed by Zeng et al. [12]. They applied the piece-wise convolutional neural networks with multi-instance learning for relationship extraction, which we refer as PCNNs. In their experiments, the dataset which contains the New York Times articles labeled with Freebase relationships was used. The sentences from 2005—2006 were used for training, while the ones from 2007 were used for testing. To compare the performance between PCNNs and our approach, we apply our approach on this dataset. As we have introduced in Section 1, Freebase only contains the major acquisitions of companies, which can be considered as the ownership of relationship. However, all of the instances of the ownership of relationship were mislabeled to be negative in the original dataset. Therefore, to compare PCNNs with our approach for extracting the ownership of relationship, we had to relabel the training set according to the corresponding Freebase relationships (99 pairs are matched in the training set). Since only 14 Freebase relationships can be matched in the test set, we randomly picked and manually validated 100 company pairs (including 50 positives and 50 negatives) from the articles in 2007. For this specific type of relationship, PCNNs labels 5 pairs as positive, which are all correct. Our approach extracts 19 pairs, where 18 of them are correct. In this experiment, our approach outperforms PCNNs in both recall and F-measure.

Regarding efficiency, our approach can extract business relationships at a rate of about 650 documents per minute on a standard consumer PC, with most of the time spent on preprocessing. The efficiency can be further improved by implementing a distributed system to apply our approach as the strategy introduced in [9].

More detailed statistics as well as the annotated data are available online.

6 Conclusion and Future Work

The focus of this work was to efficiently extract complex business relationships from news articles. We are the first to focus on the class of many-to-many relationships. To this end, we proposed a relationship extraction approach that not only extracts new relationships from text but also indicates their direction in case of non-symmetric ones, such

---

2 The original code is available online: http://www.nlpr.ia.ac.cn/cip/\-liukang/publications.html
3 http://iesl.cs.umass.edu/riedel/ecml/
as the ownership of relationship. Another contribution is the holistic pattern identification strategy, which is used to avoid the semantic drift of generated extraction patterns while dealing with multiple business relationships simultaneously.

Further, we would like to include the duration and domain information of relationships. Moreover, the performance of our approach can be further improved by understanding the semantics of the underlying sentences to avoid incorrect extractions caused by misinterpretations.

References