The Information Systems Group at HPI

Felix Naumann and Ralf Krestel
Hasso Plattner Institute
Potsdam, Germany
firstname.lastname@hpi.de

ABSTRACT

The Hasso Plattner Institute (HPI) is a private computer science institute funded by the eponymous SAP co-founder. It is affiliated with the University of Potsdam in Germany and is dedicated to research and teaching, awarding B.Sc., M.Sc., and Ph.D. degrees.

The Information Systems group was founded in 2006, currently has around ten Ph.D. students and about 15 masters students actively involved in our research activities. Our initial and still ongoing research focus has been the area of data cleansing and *duplicate detection*. More recently we have become active in the area of *text mining* to extract structured information from text, and even more recently in *data profiling*, i.e., the task of discovering various metadata and dependencies from a data instance.

1. MOTIVATION

Data abounds – it appears in many forms ranging from traditional relational or XML databases over semi-structured data, often published as linked open data, to textual data from documents on the Web. This wealth of data is ever growing, and many organizations and researchers have recognized the benefit of integrating it into larger sets of homogeneous, consistent, and clean data. Integrated data consolidates disconnected sources in organizations; it combines experimental results to gain new scientific insights; it provides consumers with a more complete view of product offers, etc.

Yet integration of such data is difficult due to its often extreme heterogeneity: Syntactic heterogeneity in data formats, access protocols, and query languages is typically the most simple to overcome, usually by building appropriate source-specific wrapper components. Next, structural heterogeneity must be overcome by aligning the different schemata of the datasets: Schema matching techniques automatically detect similarity and correspondence among schema elements, while schema mapping techniques interpret these to actually transform the data. Finally, to overcome semantic heterogeneity the different meanings of data and the similar but different representations of real-world entities must be recognized. Here, similarity search and data cleansing techniques are employed.

While the first two challenges have been research topics of our's in the past, the last and arguably most difficult challenge is a main focus of our current research endeavors. This focus manifests itself in three main research directions, which are motivated in the following sections: First, and most recently, in the area of *data profiling*, i.e., the development of methods to discover interesting properties about unknown datasets. Second, in the area of *data cleansing*, i.e., the development of methods to automatically correct errors and inconsistencies in databases and in particular to search and consolidate duplicates. Third, the area of *text mining*, i.e., the extraction of information from textual data, such as Wikipedia articles, tweets, or other text.

Where possible we aim at making our data and our algorithms available. A good starting point to find them is `http://hpi.de/naumann/projects/repeatability.html`.

2. DATA PROFILING

"Data profiling is the set of activities and processes to determine the metadata about a given dataset." [1] The need to profile a new or unfamiliar data arises in many situations, in general to prepare for some subsequent task. Data profiling comprises a broad range of methods to efficiently analyze a given dataset. In a typical scenario, mirroring the capabilities of commercial data profiling tools, tables of a relational database are scanned to derive metadata including data types and typical value patterns, completeness and uniqueness of columns, keys and foreign keys, and occasionally functional dependencies and association rules. In addition, research (ours and others') has proposed many methods for further tasks, such as the discov-

ery of inclusion dependencies or conditional functional dependencies. There are a number of concrete use cases for data profiling results, including:

- Query optimization: counts and histograms for selectivity estimation, dependencies for query simplification
- Data cleansing: pattern and dependency detection to identify violations
- Data integration: inter-database inclusion dependencies to enrich datasets and find join-paths
- Data analytics: data preparation and initial insights
- Database reverse engineering: foreign key discovery to understand a schema and identify its core components

Our survey [1] highlights the community's significant research progress in this area in the recent past. Data profiling is becoming a more and more popular topic as researchers and practitioners are recognizing that just gathering data into data lakes is not sufficient: "If we just have a bunch of data sets in a repository, it is unlikely anyone will ever be able to find, let alone reuse, any of this data. With adequate metadata, there is some hope [...]" [4]

2.1 Profiling relational data

Apart from computationally more simple tasks, such as counting the number of distinct values in a column, data profiling is typically concerned with discovering dependencies in a given, possibly large dataset. We, and other groups, have developed various methods to efficiently discover all minimal functional dependencies, inclusion dependencies, unique column combinations, and order dependencies. More dependencies are to come, such as join dependencies, matching dependencies, denial constraints, etc. Instead of listing and explaining each technique in any detail, we highlight some general difficulties we have encountered that make data profiling both challenging and interesting:

Schema size: Because dependencies can occur among any column or column combination, not only the number of records, but also the number of columns is a decisive factor of complexity.

Size of dependencies: One way to handle the exponential search space is to limit the size of the dependencies, i.e., the number of involved attributes. For instance, one could argue that key-candidates with more than ten attributes are not useful. On the other hand, a complete set of metadata can be useful, for instance to normalize a relation based on its functional dependencies.

Number of dependencies: While much dependency-focussed research, such as normalization theory or reasoning with dependencies, assumes a handful of dependencies as input, we typically observe thousands, millions and in some cases even billions of dependencies in typical real-world datasets. Just storing them becomes a problem, not to mention reasoning about them or interpreting them manually.

Treatment of nulls: The semantics of missing values is an interesting problem for almost any data management and analysis task, likewise for data profiling [13].

Intricate pruning: Huhtala et al. already showed quite complex insights to efficiently prune the search space for FD discovery [10]. When profiling for various types of dependencies, cross-dependency pruning becomes possible.

Relaxed dependencies: Apart from strict dependencies, it is also of interest to discover partial dependencies, which are true for only a part of the dataset, and conditional dependencies, which are true for a well-defined such part.

Dynamic data: While most of our focus has been on algorithms for a given, static dataset, we are also interested in efficiently updating data profiling results after changes in the data.

Experiments: Testing correctness of algorithms for given, real-world datasets is straightforward, but generating artificial testdata with certain properties, such as a certain number and distribution of functional dependencies, is very challenging.

Interpreting results: Any discovered metadata can only be validated for the dataset at hand. Some might be true in general, some might be spurious. We discuss this arguably most important and most difficult challenge of making sense of profiling results in Section 2.4.

In conclusion, research has many avenues to follow!

2.2 The Metanome project

Metanome is our open Java-based framework and tool for managing relational datasets and data profiling algorithms [18]. Our motivation for this undertaking is to bundle the many algorithms developed in our group, to provide an easy interface and testing environment for developers of new algorithms, and finally to enable fair comparisons among competing algorithms. Our initial focus was on functional dependency discovery, and Metanome features implementations of already eight published FD-discovery algorithms including those evaluated in [19] plus seven further algorithms for other discovery tasks (`www.metanome.de`).

2.3 Profiling RDF data

Among the datasets that are particularly worthy to profile, due to their variety and their general interest, are linked datasets. We are applying traditional and novel data mining technology to linked data in its RDF representation as subject-predicate-object triples. For instance, the discovery of frequent itemsets of predicates or objects in the context of subjects allows enriching datasets with missing triples. Another configuration – mining for frequent subjects in the context of predicates – achieves a clustering of entities. We have also applied data mining techniques for the discovery of conditional inclusion dependencies [16]. The volume of available linked data (a popular dataset is from the Billion Triples Challenge, which currently comprises over 3 billion facts) necessitates space-efficient algorithms.

Again, much of our work enters our browser-based discovery tool, ProLOD++ [2], which features techniques to discovery key-candidates, explore class and property distributed, discover frequent graph patterns, and more (see Figure 1).

2.4 From metadata to semantics

Finding all (and thus very many) dependencies in a given dataset is only the first part of a meaningful discovery process. The vast majority of metadata is spurious: It might be valid only in the current instance, or it might be valid for any reasonable instance but meaningless nonetheless. Separating the wheat from the chaff is extremely difficult, as it is a jump from (meta-)data to semantics; only a human can promote a unique column combination to a key, an inclusion dependency to a foreign key, or a functional dependency to an enforced constraint.

But computer science can help: We are currently investing much of our time to transform large amounts of metadata to schematic information. A first step is a metadata management system to store and query many different types of metadata. Next, we are developing selection and ranking methods to present to users only the most promising metadata. And finally, the visualization of metadata is an important tool to aid experts in understanding their data. Figure 2, for instance, shows connected components created by discovering inclusion dependencies among millions of web tables.

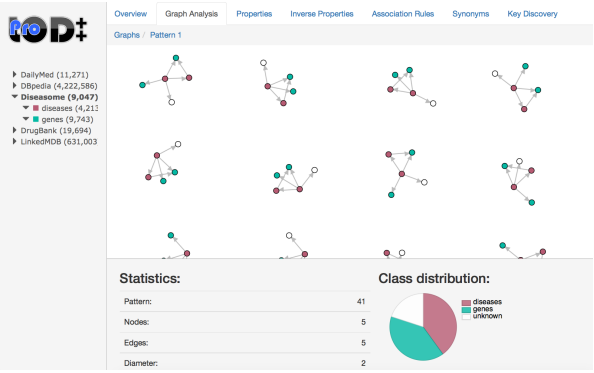

Figure 1: Exploring frequent patterns in a Linked Dataset with ProLOD (`www.prolod.org`)

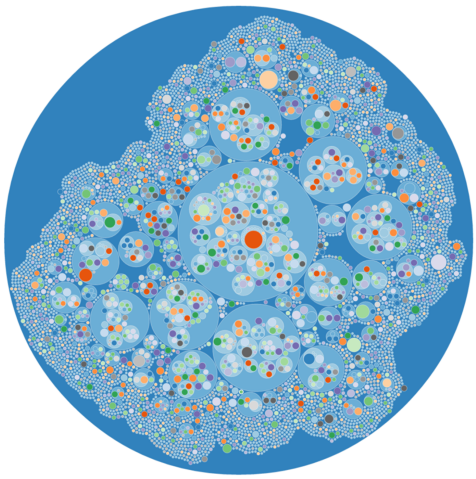

Figure 2: Clusters of web tables, connected through (reasonable) inclusion dependencies

3. DATA CLEANSING

With the ever-increasing volume of data, data quality problems arise. One of the most intriguing problems is that of multiple, yet different representations of the same real-world object in the data: duplicates. Such duplicates have many detrimental effects, for instance bank customers can obtain duplicate identities, inventory levels are monitored in-

correctly, catalogs are mailed multiple times to the same household, etc. A related problem is that of similarity search in structured data: given a query record, find the most similar candidate records in a database and identify whether one of them is a match.

The areas of similarity search and duplicate detection are experiencing a renaissance both in research and industry. Apart from scientific contributions we cooperate with companies to transfer our technology. Both our similarity search and our duplicate detection techniques have been adopted by industry partners.

3.1 Duplicate detection

Detecting duplicates is difficult: First, duplicate representations are usually not identical but slightly differ in their values. Second, in principle, all pairs of records should be compared, which is infeasible for large volumes of data [9]. Our research addresses both aspects by designing effective similarity measures and by developing efficient algorithms to reduce the search space.

One focus of our work is to develop improved variations of the elegant and simple sorted neighborhood method [8], for instance adapting it to nested XML data, making it progressive, parallelizing it for GPU-processing, or creating an adaptive version that is provably more efficient than the original [5].

In our experience, research(ers) in duplicate detection suffers particularly when trying to transfer technology and methods to industrial settings: Availability of data is a first issue, that arises even if a cooperation is firmly established and all participating parties in principle agree to the effort. Next, domain- and partner-specific similarity measures are needed that satisfy the specific use-case. Companies can have widely differing views of what constitutes a duplicate: Measuring recall is impossible due to large dataset sizes, and precision is surprisingly malleable, depending on whom one asks for validation. And finally, the real world holds many nitty, gritty details that can be conveniently ignored in a research setting[1]. With [20] we were able to overcome these difficulties and have had a lasting impact on the data quality of our partner.

3.2 Similarity search

A problem related to duplicate detection, but offering quite different requirements is that of efficient similarity search. Instead of comparing all or many pairs of records in an offline fashion ($n \times n$), online similarity search asks for all records matching a given query record ($1 \times n$). A typical use case is a call center agent pulling up customer information based on a customer's name and city. The main challenge is to develop a suitable similarity index, a much more difficult undertaking that an exact-match index.

One of our solutions matches the problem to a query plan optimization task, choosing similarity index accesses based on their selectivity and their cost, each of which is again modified by the dynamically chosen threshold: A low thresholds yields more candidates, but also more access to disk to retrieve the candidates [17]. A further insight is the importance of frequency-aware similarity measures, which apply different weights depending on the frequency of the query terms (Schwarzenegger vs. Miller).

We are currently extending this work to solve the problem of an ever-growing set of data that shall be held duplicate free: each query can simultaneously be an insert-operation.

[1]For instance, providing a machine with 16GB main memory but insisting on a 32-bit operating system.

4. TEXT MINING

Unstructured data in the form of textual documents can be found everywhere, from medical records to game chats, and from politicians' speeches to tweets. These documents cover a variety of genres, from serious to fun, from entire novels to single words. This diversity makes dealing with textual data particularly challenging and there is no one-size-fits-all text mining method yet. We are currently working on the topics of named entity linking, topic modeling, and bias detection on various document collections from the web. We cover the research areas natural language processing, information extraction, and recommender systems.

4.1 Named entity linking

A first step in analyzing texts is to find entities. Named entity linking is a rather new task composed of named entity recognition and linking the textual mentions in a document to corresponding entries in a knowledge base, thus disambiguating the mentions. The disambiguation can be performed using additional information in the knowledge base and the context of the mentions in the documents. We developed a named entity linking approach that operates on a textual range of relevant terms. We then aggregate decisions from an ensemble of simple classifiers, each of which operates on a randomly sampled subset from the above range [21]. The obtained results are very good with respect to precision and recall.

Some tasks, such as topic-based clustering, require near perfect precision and therefore we enhanced our named entity linking approach using random walks [6]. This allows for efficient computation of the linking and improves the precision at a minimal expense of recall.

4.2 Relationship extraction

Once entities are successfully extracted and disambiguated, finding relations between those entities is a next logical step. In the context of an industry project with a large German bank, we aim at building company networks to support their risk management department. These company networks are extracted automatically from newspaper articles, posing new challenges to the named entity recognition task, which is particularly difficult for German company names, due to complex, often ambiguous naming. Further, the relationship types we are interested in differ from standard, binary relations, such as "married with" or "located in". Our company networks require the detection of relations that are not necessarily binary, e.g. "competitor with" or "supplier to". To this end, we developed a holistic, seed-based algorithm to find these types of relations by providing a handful of example instantiations. The algorithm is based on Snowball [3] and can deal with any type of user-provided relations to extract relationship types with high precision.

4.3 Recommender systems

As with the information extraction tasks, we have a strong focus on the application of our research. Therefore, personalization, prediction, and recommendation play a major role in our group's work. From predicting accepted answers in MOOC forums [11], to recommending hashtags in Twitter [7], we analyzed diverse text collections accessible through the web. We also experimented with recommending serendipitous news articles [12] to present to the user not only relevant and novel articles, but also some surprising ones.

In an attempt to bridge the gap between traditional news and social media, we developed a tweet recommender system [15]. The goal was to provide the reader of a news article about some event with an overview of the reactions in Twitter. While Twitter is often only used to share and distribute information, it is also used to express opinions, reject ideas, or support certain viewpoints. To detect these (subtle) opinions, traditional sentiment analysis techniques have to be adapted to recognize emojis, abbreviations, slang, etc. The mismatch between the language used in news articles and tweets makes recommending one based on the other challenging.

4.4 Bias detection

Finally, we have to deal with another mismatch between used languages when trying to detect political bias of mainstream media. Initial experiments on comparing parliamentary speeches with news articles of various German news outlets [14] have shown that perceived bias can be automatically quantified. Given the very different genres (speeches vs. articles), detecting biased statements based on their comparison is rather cumbersome. Only rarely vocabulary use is a good indicator (e.g., "nuclear energy" vs. "atomic energy" in Germany). Nevertheless, identifying this bias in mainstream media and making it visible to the reader is an important piece of information.

Beside this *statement bias*, newspapers can also influence their readers by only reporting about certain topics (*gate-keeping bias*) or covering certain positions more thoroughly than others (*coverage bias*). Automatically detecting all three kinds of bias is our current goal, making it necessary to extract not only entities (politicians, parties, domain experts) and their relations, but also to do fine-grained opinion mining and sentiment analysis.

5. ACKNOWLEDGMENTS.

Our research has been funded by various partners including the DFG and companies interested in understanding and improving their data. We are very grateful for being able to work with our great Ph.D. students, who currently are Toni Gruetze, Hazar Harmouch, Maximilian Jenders, Anja Jentzsch, John Koumarelas, Sebastian Kruse, Konstantina Lazaridou, Michael Loster, Thorsten Papenbrock, Ahmad Samiei, and Zhe Zuo.

Two further groups at HPI, with which we collaborate, are based in the database community: The Enterprise Platforms and Integration Concepts (EPIC) group headed by Hasso Plattner and Matthias Uflacker, and the Knowledge Discovery and Data Mining (KDD) group headed by Emmanuel Müller.

6. REFERENCES

[1] Z. Abedjan, L. Golab, and F. Naumann. Profiling relational data: a survey. *VLDB Journal*, 24(4):557–581, 2015.

[2] Z. Abedjan, T. Grütze, A. Jentzsch, and F. Naumann. Profiling and mining RDF data with ProLOD++. In *Proceedings of the*

International Conference on Data Engineering (ICDE), 2014. Demo.

[3] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital Libraries*, pages 85–94, 2000.

[4] D. Agrawal et al. Challenges and opportunities with Big Data. Technical report, Computing Community Consortium, `http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf`, 2012.

[5] U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg. Adaptive windows for duplicate detection. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1073–1083, Washington, D.C., 2012.

[6] T. Gruetze, G. Kasneci, Z. Zuo, and F. Naumann. CohEEL: Coherent and efficient named entity linking through random walks. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2016.

[7] T. Gruetze, G. Yao, and R. Krestel. Learning temporal tagging behaviour. In *Proceedings of the Temporal Web Analytics Workshop (TempWeb) at the International World Wide Web Conference (WWW)*, 2015.

[8] M. A. Hernández and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.

[9] M. Herschel, F. Naumann, S. Szott, and M. Taubert. Scalable iterative graph duplicate detection. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 24(11), 2012.

[10] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate dependencies. *Computer Journal*, 42(2):100–111, 1999.

[11] M. Jenders, R. Krestel, and F. Naumann. Which answer is best? Predicting accepted answers in MOOC forums. In *WWW Companion*, 2016.

[12] M. Jenders, T. Lindhauer, G. Kasneci, R. Krestel, and F. Naumann. A serendipity model for news recommendation. In *Advances in Artificial Intelligence - Annual German Conference on AI (KI)*, volume 9324 of *Lecture Notes in Computer Science*, pages 111–123, 2015.

[13] H. Köhler, S. Link, and X. Zhou. Possible and certain SQL keys. *Proceedings of the VLDB Endowment*, 8(11):1118–1129, 2015.

[14] R. Krestel, A. Wall, and W. Nejdl. Treehugger or Petrolhead? Identifying Bias by Comparing Online News Articles with Political Speeches. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 547–548, 2012.

[15] R. Krestel, T. Werkmeister, T. P. Wiradarma, and G. Kasneci. Tweet-recommender: Finding relevant tweets for news articles. In *Proceedings of the International World Wide Web Conference (WWW)*, 5 2015.

[16] S. Kruse, A. Jentzsch, T. Papenbrock, Z. Kaoudi, J.-A. Quiane-Ruiz, and F. Naumann. RDFind: Scalable conditional inclusion dependency discovery in RDF datasets. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, 2016.

[17] D. Lange and F. Naumann. Efficient similarity search: Arbitrary similarity measures, arbitrary composition. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, Glasgow, UK, 2011.

[18] T. Papenbrock, T. Bergmann, M. Finke, J. Zwiener, and F. Naumann. Data profiling with metanome (demo). *Proceedings of the VLDB Endowment*, 8(12):1860–1871, 2015.

[19] T. Papenbrock, J. Ehrlich, J. Marten, T. Neubert, J.-P. Rudolph, M. Schnberg, J. Zwiener, and F. Naumann. Functional dependency discovery: An experimental evaluation of seven algorithms. *Proceedings of the VLDB Endowment*, 8(10):1082–1093, 2015.

[20] M. Weis, F. Naumann, U. Jehle, J. Lufter, and H. Schuster. Industry-scale duplicate detection. *Proceedings of the VLDB Endowment*, 1(2):1253–1264, 2008.

[21] Z. Zuo, G. Kasneci, T. Gruetze, and F. Naumann. BEL: Bagging for entity linking. In *International Conference on Computational Linguistics (COLING)*, 2014.