CurEx – A System for Extracting, Curating, and Exploring Domain-Specific Knowledge Graphs from Text

Michael Loster, Felix Naumann
Hasso Plattner Institute
University of Potsdam
firstname.lastname@hpi.de

Jan Ehmueller, Benjamin Feldmann
Hasso Plattner Institute
University of Potsdam
firstname.lastname@student.hpi.de

ABSTRACT

The integration of diverse structured and unstructured information sources into a unified, domain-specific knowledge base is an important task in many areas. A well-maintained knowledge base enables data analysis in complex scenarios, such as risk analysis in the financial sector or investigating large data leaks, such as the Paradise or Panama papers. Both the creation of such knowledge bases, as well as their continuous maintenance and curation involves many complex tasks and considerable manual effort.

With CurEx, we present a modular system that allows structured and unstructured data sources to be integrated into a domain-specific knowledge base. In particular, we (i) enable the incremental improvement of each individual integration component; (ii) enable the selective generation of multiple knowledge graphs from the information contained in the knowledge base; and (iii) provide two distinct user interfaces tailored to the needs of data engineers and end-users respectively. The former has curation capabilities and controls the integration process, whereas the latter focuses on the exploration of the generated knowledge graph.

ACM Reference Format:
Michael Loster, Felix Naumann and Jan Ehmueller, Benjamin Feldmann. 2018. CurEx – A System for Extracting, Curating, and Exploring Domain-Specific Knowledge Graphs from Text. In *Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM '18).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269229

1 INTRODUCTION

Knowledge bases, created from both structured and unstructured information sources, are a corner stone of many advanced systems. They enable the analysis of complex relationships that would be difficult to conduct without structured knowledge. As an example, modern risk analysis techniques can greatly benefit from an integrated knowledge base, especially as they can be used to create dedicated domain-specific knowledge graphs. These graphs can in turn be used to gain a holistic view of the current economic situation, so that systemic risk factors can be identified early enough to react appropriately. Other financial scenarios, such as investigating the impact of corporate bankruptcy on other market participants, can also leverage knowledge graphs to gain valuable insights. Here, the links between the individual market participants can be used to determine which companies are affected by a bankruptcy and to what extent. In data journalism, knowledge bases that contain documents from large data leaks, such as the Panama or Paradise papers, help to discover and unravel complex network structures that can be indicative of tax evasion. By taking a temporal dimension into account, it is also possible to examine the effects of structural changes over a certain period of time. The insights gained can thus be used to quickly identify critical constellations and counteract their effects.

The integration of structured and unstructured data sources has a long-standing history [4] in which many systems with different specialisations have emerged. Some of the more recent ones include Data Tamer [6], Deep Dive [5] and BigGorilla [1]. Creating a system capable of automatically generating a domain-specific knowledge base, which can then be used to derive knowledge graphs, involves many well-known challenges that need to be addressed. In order to uniquely identify each entry in the knowledge base, no duplicate entries should be created during the integration process. To avoid this, entities that refer to the same real-world entity must be identified and merged into a single entity. This entity resolution process poses a great challenge, since many entities cannot be assigned unambiguously and are therefore difficult to resolve in an automatic manner. Another challenge is the extraction, as well as the subsequent integration of information originating from unstructured data sources, such as newspaper articles or annual reports. For an orderly integration of such information, it must first be extracted from the respective data sources using extraction methods such as Named Entity Recognition (NER) and Relation Extraction (RELEX). Subsequently, the obtained information can be incorporated into the knowledge base, by employing entity linking (EL) methods, such as CohEEL [2]. Because each of these steps involves a certain level of error, it is necessary for a real-world system to introduce adequate analysis, monitoring, and curation capabilities to control and ensure the quality of the resulting knowledge base.

To meet these challenges, we present CurEx, a system that is capable of creating and maintaining a knowledge graph consisting of various named entities and their relationships. The demonstrated system generates a knowledge graph by extracting and combining information from both structured data sources, such as Wikidata and DBpedia, as well as from unstructured data sources, such as newspaper articles. The integrated knowledge base ultimately comprises almost 2.1 million entities and roughly 18 million relationships, including co-occurrences. To this end, CurEx is able to recognize named entities in structured and unstructured sources, link them with the information of a knowledge base, and extract

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269229

the relations expressed in the text between the identified entities. The constructed knowledge base is then used to generate a domain-specific knowledge graph. To achieve this, our system consists of several components. Each of them addresses a specific subproblem and can be easily exchanged due to the modular system architecture in order to benefit from future advancements in the respective research areas.

2 SYSTEM OVERVIEW

Since the goal of CurEx is to integrate information from structured and unstructured data sources, it is divided into two main components that are specifically tailored to the integration of the respective data sources. Figure 1 shows a general overview of the system. It is entirely based on scalable technologies, such as Apache Spark and Cassandra and is therefore able to process large amounts of data. Another important system aspect is its modular architecture. This modularity is achieved by implementing all components as Spark jobs and thus making them easy to replace. While the Integration component focuses on integrating structured data sources, the Text Mining component handles the information extraction from unstructured data sources as well as their subsequent integration into the knowledge base. As can be seen in Figure 1, the Text Mining component is divided into three subcomponents: Named Entity Recognition (NER), Entity Linking (EL) and Relation Extraction (RELEX). These three components are responsible for identifying named entities, link them to the information in the knowledge base, and extract the relationships between the discovered entities from the text. The resulting knowledge base is then used to create a knowledge graph of named entities and their relations. To access the information contained in the knowledge graph, it is exported into a graph database, such as Neo4j. This export allows the selective creation of knowledge graphs based only on certain parts of the knowledge base. This requirement might be necessary due to security clearances or other restrictive criteria.

CurEx introduces two distinct user interfaces customized for the needs of data engineers and of end-users. Data engineers can use the Curation Interface to not only control and monitor many steps of the integration process, but also to directly make changes to the generated knowledge base. End-users can use the Entity Landscape Explorer (ELEX) to explore the resulting knowledge graph and provide valuable feedback to the data engineers.

2.1 Structured Data Integration

The purpose of the Integration component is to integrate multiple structured data sources, merging all entities that refer to the same real-world entity. This component is divided into two subcomponents, one for normalizing and one for deduplicating the datasets. The following structured data sources were used to create an initial knowledge base: the German Wikidata, the German DBpedia, Implisense, and Kompass. The latter two data sources are manually curated, commercially available company datasets.

Normalization. Since in many cases the format of the attribute values differs between data sources, semantically equivalent attribute names and their contents must first be normalized. To give an example, some sources refer to geo-coordinates, as "coordinate_location", others simply as "geo.coords". During the normalization step, attributes with semantically identical information are not only named uniformly, but also merged if required.

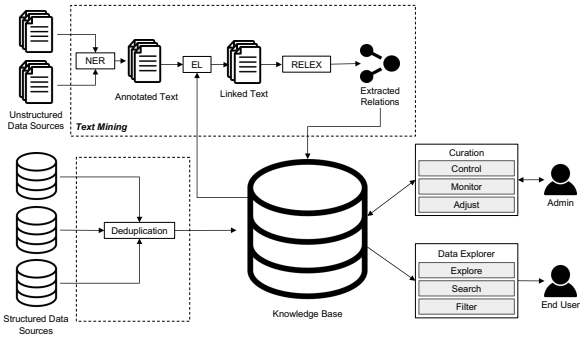

Figure 1: System Architecture

Deduplication. As a starting point for the development of our knowledge base we use the manually maintained Implisense dataset of an industry partner and merge it step by step with all other structured data sources. To create a duplicate-free knowledge base, entities that refer to the same real-world object must be merged during the deduplication process. To minimize comparisons, we utilize a blocking technique that essentially groups similar objects into smaller blocks according to a specific blocking key (e.g., zip code). In this way, a quadratic comparison can be carried out within each individual block.

For the actual detection of duplicate entries, the system is able to compare any subset of attributes such as names, unique identifiers or URLs. We employ different similarity measures to first determine the similarity of individual attribute pairs and then combine the resulting similarity values using a simple linear combination. Finally, we normalize the calculated value to the range [0,1], and use the resulting value to decide whether two entities are duplicates or not. Overall, the resulting knowledge base consists of 2.1 million entities, with approximately 29,000 entities occurring in exactly two and 3,700 entities occurring in exactly three sources.

2.2 Text Mining

After creating an initial knowledge base, we extend it with relationships extracted from unstructured texts. To this end, we use several text mining techniques for the recognition of named entities and the extraction of relationships between them. This task is carried out by the Text Mining component, which consists of three submodules, namely Named Entity Recognition (NER), Entity Linking (EL) and Relation Extraction (RELEX). Each of them is briefly outlined below.

Named Entity Recognition. The NER component is used for the discovery of named entities in unstructured texts. Here we use an approach similar to Loster et al. [3], which first creates large dictionaries of externally available knowledge and then integrates it into the training process of a conditional random field classifier.

Entity Linking. Because an entity extracted by the NER component might refer to many possible entries in the knowledge base, it becomes necessary to disambiguate and link these entities with their corresponding entry in the knowledge base. This component currently relies on a fuzzy matching approach, which employs different string similarities to detect identical entities. However, we use the more sophisticated CohEEL approach [2] to discover the links between the knowledge base entries and the German Wikipedia articles. Since this approach, also considers the context of a company mention, it allows us to find the best of several possible knowledge base matches. This turns out to be particularly useful for linking companies that are operating in different sectors and have very similar names, as their textual context can often be used as an indicator for a unique match.

Relation Extraction. Finally, it is the focus of the relation extraction (RELEX) component to detect relationships between the previously discovered entities. The component currently in use extracts co-occurrence relationships between individual entities found within the same sentence. Due to the modular system architecture, it is possible to replace the currently used RELEX module with more advanced extraction techniques, such as the technique presented by Zuo et al. [8] or even more advanced neural network based techniques [7]. In particular, the approach proposed by Zuo et al. is promising, as it is able to extract directional relationships where the arguments are of the same type, for example, company to company relationships. Since the entities were already linked to the knowledge base in the previous step, the extracted relationships can easily be integrated into the knowledge base.

3 INTERFACE & INTERACTIONS

Every real integration system needs to adhere to the requirements of different user groups. For example, data engineers must be able to control the individual steps of the integration process and directly adopt the knowledge base, while end users of the integrated data are more interested in methods for the efficient exploration of the data. To this end, we propose two separate user interfaces, specifically tailored to the needs of two user groups, namely data engineers and end users. The Entity Landscape Explorer is designed for the end-user and allows him/her to efficiently explore and inspect the knowledge base by means of browsing through the generated knowledge graph. The Curation Interface, on the other hand, is aimed at the data engineers and enables them to control the integration process and to curate the knowledge base.

3.1 Entity Landscape Explorer

The Entity Landscape Explorer (ELEX) shown in Figure 2 is designed to meet the needs of the end user. Since a knowledge graph can easily contain thousands of nodes and edges, a user must be able to efficiently examine the graph and the associated knowledge base. ELEX provides this functionality by allowing the user to explore the graph using appropriate filtering, layout and search methods. The starting point for such a focused exploration is an initial node, which can be selected by the user through a search interface. To focus the exploration even more, the user can limit the exploration to nodes of a certain entity type as well as to certain edge types. When examining a specific subgraph, nodes and edges can be filtered out dynamically, making it possible to reveal structures that were difficult to recognize beforehand. For a better overview, the currently displayed nodes can also be arranged in a tree structure, which is particularly useful when analyzing customer-supplier relationships. To thin out the graph even further, it is possible to completely show or hide nodes and edges of a certain type. Another key feature of ELEX is the ability to examine the individual attributes of each node and edge. Since the information stored in the knowledge base is not necessarily error-free, it is an important part of ELEX to give the user the opportunity to correct erroneous entries through a feedback mechanism.

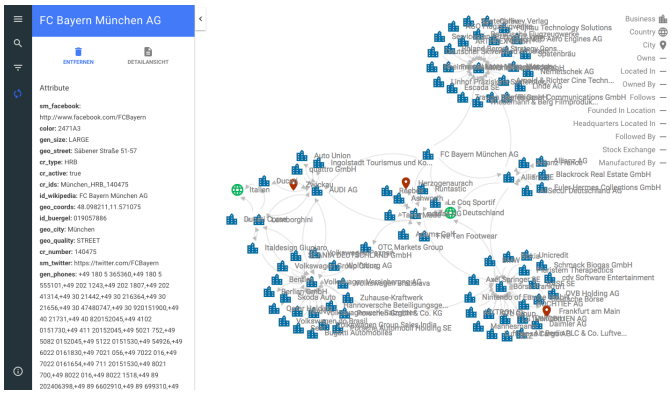

Figure 2: Entity Landscape Explorer

3.2 Curation Interface

The functionality of the Curation Interface shown in Figure 3 is tailored to the needs of data engineers. This user group is interested in controlling the actual integration process, inspecting the resulting knowledge base, and if necessary, making changes to improve information quality. For one, it must be possible to minimize the errors caused by the successive data transformation steps, and also, it is vital to permit the correction of errors that already existed in the original data sources. The Curation Interface was developed with these core aspects in mind and offers appropriate functionalities to address them. As can be seen in Figure 3, the interface provides several tabs, which in turn are divided into different groups. The "Subjects", "Versions", and "Graphs" tabs are used to curate the knowledge base. Among other things, they allow the user to make changes directly in the knowledge base, roll back the knowledge base to an earlier state, and edit the business entities in a graph view, which simplifies the handling of relationships. The "Duplicates", "Blocking Statistics", and "Similarity Measure" tabs on the other hand are used to display and evaluate the results of the deduplication component. These tabs show the results of a duplication run between the knowledge base and a new data source, provide blocking statistics to estimate the cluster utilization and show precision, recall and F1 measure of the currently used similarity measure in case a gold standard is available for a subset of the data. Finally, the text mining group consists of the "Entity Linking" and "Classifier Statistics" tabs, which make it possible to view the results of the entity linking subcomponent and to evaluate various classification models.

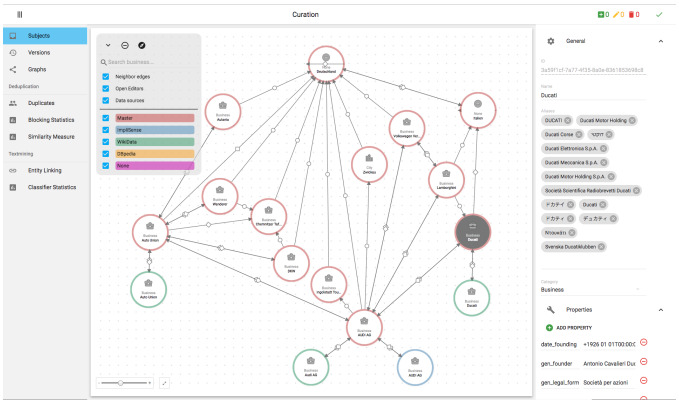

Figure 3: Graph view in the Curation Interface

4 DEMONSTRATION OVERVIEW

First we present the curation of the knowledge base including the monitoring of the intermediate data processing steps. Afterwards we demonstrate the graph exploration using ELEX. During the demonstration we will guide the participants through the following steps:

(1) **Data inspection**: We start by giving an overview of the business entities in the knowledge base. We search for a specific business entity and present various possibilities to present the available information. We then show the subgraph around the selected business entity and demonstrate the difference between the conventional tabular view and the graph view of the entity's attributes and relations.
(2) **Data curation**: We demonstrate the curation capabilities of the graph view, by changing the attributes of a selected entity. In addition, we add new business entities and delete existing ones. We then present the status view of the changes made. The operations displayed will be applied to the knowledge base in the next step of the demonstration.
(3) **History overview**: During this part of the demonstration we show the various data processing steps that have been applied to the knowledge base. As a starting point, we show the difference between the current state of the knowledge base and the knowledge base before manual curation. We then commit the changes made in the previous step to the knowledge base.
(4) **Monitoring of intermediate steps**: We present the tools for monitoring the intermediate steps of the Deduplication and Text Mining components. We first show the duplicates of a previously performed duplication run and inspect some of them. Then we show the evaluation of different blocking keys as well as the evaluation of different threshold values for the duplicate detection. For the Text Mining component we will first demonstrate the entity linking overview. This view displays both articles and their links to the corresponding knowledge base entires. We then show the evaluation tool for the different classification models used within the Text Mining component. Participants are presented with several evaluations for different classification models and their parameters.

After demonstrating the Curation Interface, we continue the demonstration by introducing ELEX. We first show that the changes made in the Curation Interface will already be accessible from ELEX. We continue the demonstration by presenting the following key features of ELEX: high-performance display and exploration capabilities of the knowledge graph, searching for single entities and filtering the knowledge graph, inspecting single entities, displaying the graph in different layouts and giving feedback to point out errors for single nodes and edges, e.g., an error in a data field or an incorrect relation.

5 CONCLUSION

We present CurEx, a modular system to integrate structured and unstructured data sources into a domain-specific knowledge base and create explorable knowledge graphs for specific domains. The system is based on scalable technologies and is therefore able to process large amounts of data, making it suitable for real-world scenarios. It consists of two major components specifically designed for integrating information from structured and unstructured data sources. Since all subcomponents are implemented as Spark jobs, they are easily replaced or extended. It provides two distinct user interfaces, each addressing the individual needs of a specific user group. As such, a data engineer can control and manage the integration process using the Curation Interface, whereas a normal end-user uses ELEX to explore the knowledge graph and submit feedback.

As future directions we focus on the improvement of the individual subcomponents. For example, we plan to replace the currently used deduplication approach with a novel approach based on neural networks. Another planned improvement is entirely replacing the fuzzy matching approach for entity linking with CohEEL.

REFERENCES

[1] Chen Chen, Behzad Golshan, Alon Y. Halevy, Wang-Chiew Tan, and AnHai Doan. 2018. BigGorilla: An Open-Source Ecosystem for Data Preparation and Integration. *IEEE Data Engineering Bulletin Issues* 41 (2018), 10–22.
[2] Toni Grütze, Gjergji Kasneci, Zhe Zuo, and Felix Naumann. 2016. CohEEL: Coherent and efficient named entity linking through random walks. *Journal of Web Semantics* 37–38 (2016), 75–89.
[3] Michael Loster, Zhe Zuo, Felix Naumann, Oliver Maspfuhl, and Dirk Thomas. 2017. Improving Company Recognition from Unstructured Text by using Dictionaries. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*. 610–619.
[4] Renee Miller. 2018. Open Data Integration. *Proceedings of the International Conference on Very Large Databases (VLDB)* 11 (2018), 2130–2139.
[5] Christopher De Sa, Alexander Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. 2017. Incremental knowledge base construction using DeepDive. *Proceedings of the International Conference on Very Large Databases (VLDB)* 26 (2017), 81–105.
[6] Michael Stonebraker, Daniel Bruckner, Ihab F. Ilyas, George Beskales, Mitch Cherniack, Stanley B. Zdonik, Alexander Pagan, and Shan Xu. 2013. Data Curation at Scale: The Data Tamer System. In *Conference on Innovative Data Systems Research CIDR*.
[7] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 207–2013.
[8] Zhe Zuo, Michael Loster, Ralf Krestel Krestel, and Felix Naumann. 2017. Uncovering Business Relationships: Context-sensitive Relationship Extraction for Difficult Relationship Types. In *Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen" (LWDA)*. 271–283.