Extraction and Representation of Financial Entities from Text

Tim Repke and Ralf Krestel

Abstract In our modern society, almost all events, processes, and decisions in a corporation are documented by internal written communication, legal filings, or business and financial news. The valuable knowledge in such collections is not directly accessible by computers as they mostly consist of unstructured text. This chapter provides an overview of corpora commonly used in research and highlights related work and state-of-the-art approaches to extract and represent financial entities and relations.

The second part of this chapter considers applications based on knowledge graphs of automatically extracted facts. Traditional information retrieval systems typically require the user to have prior knowledge of the data. Suitable visualization techniques can overcome this requirement and enable users to explore large sets of documents. Furthermore, data mining techniques can be used to enrich or filter knowledge graphs. This information can augment source documents and guide exploration processes. Systems for document exploration are tailored to specific tasks, such as investigative work in audits or legal discovery, monitoring compliance, or providing information in a retrieval system to support decisions.

1 Introduction

Data is frequently called the oil of the twenty-first century.[1] Substantial amounts of data are produced by our modern society each day and stored in big data centers. However, the actual value is only generated through statistical analyses and data mining. Computer algorithms require numerical and structured data,

[1] E.g., https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

T. Repke (✉) · R. Krestel
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
e-mail: tim.repke@hpi.de; ralf.krestel@hpi.de

S. Consoli et al. (eds.), *Data Science for Economics and Finance*,
https://doi.org/10.1007/978-3-030-66891-4_11

such as in relational databases. Texts and other unstructured data contain a lot of information that is not readily accessible in a machine-readable way. With the help of *text mining*, computers can process large corpora of text. Modern *natural language processing* (NLP) methods can be used to extract structured data from text, such as mentions of companies and their relationships. This chapter outlines the fundamental steps necessary to construct a *knowledge graph* (KG) with all the extracted information. Furthermore, we will highlight specific state-of-the-art techniques to further enrich and utilize such a knowledge graph. We will also present text mining techniques that provide numerical representations of text for structured semantic analysis.

Many applications greatly benefit from an integrated resource for information in exploratory use cases and analytical tasks. For example, journalists investigating the Panama Papers needed to untangle and sort through vast amounts of data, search entities, and visualize found patterns hidden in the large and very heterogeneous leaked set of documents and files [10]. Similar datasets are of interest for data journalists in general or in the context of computational forensics [19, 13]. Auditing firms and law enforcement need to sift through massive amounts of data to gather evidence of criminal activity, often involving communication networks and documents [28]. Current computer-aided exploration tools,[2] offer a wide range of features from data ingestion, exploration, analysis, to visualization. This way, users can quickly navigate the underlying data based on extracted attributes, which would otherwise be infeasible due to the often large amount of heterogeneous data.

There are many ways to represent unstructured text in a machine-readable format. In general, the goal is to reduce the amount of information to provide humans an overview and enable the generation of new insights. One such representation are *knowledge graphs*. They encode facts and information by having nodes and edges connecting these nodes forming a graph.[3] In our context, we will consider nodes in the graph as named entities, such as people or companies, and edges as their relationships. This representation allows humans to explore and query the data on an abstracted level and run complex analyses. In economics and finance, this offers access to additional data sources. Whereas internally stored transactions or balance sheets at a bank only provide a limited view of the market, information hidden in news, reports, or other textual data may offer a more global perspective.

For example, the context in which data was extracted can be a valuable additional source of information that can be stored alongside the data in the knowledge graph. Topic models [8] can be applied to identify distinct groups of words that best describe the key topics in a corpus. In recent years, embeddings significantly gained popularity for a wide range of applications [64]. Embeddings represent a piece of text as a high-dimensional vector. The distance between vectors in such a vector space can be interpreted as semantic distance and reveals interesting relationships.

[2] E.g., extraction and indexing engine (https://www.nuix.com/), network analysis and visualization (https://linkurio.us/), or patent landscapes (https://clarivate.com/derwent/).

[3] Knowledge graphs are knowledge bases whose knowledge is organized as a graph.

This chapter focuses on the construction and application of knowledge graphs, particularly on company networks. In the first part, we describe an NLP pipeline's key steps to construct (see Sect. 2) and refine (see Sect. 3) such a knowledge graph. In the second part, we focus on applications based on knowledge graphs. We differentiate them into syntactic and semantic exploration. The syntactic exploration (see Sect. 5) considers applications that directly operate on the knowledge graph's structure and meta-data. Typical use cases assume some prior knowledge of the data and support the user by retrieving and arranging the relevant extracted information. In Sect. 6 we extend this further to the analogy of semantic maps for interactive visual exploration. Whereas syntactic applications follow a localized bottom-up approach for the interactions, semantic exploration usually enables a top-down exploration, starting from a condensed global overview of all the data.

2 Extracting Knowledge Graphs from Text

Many business insights are hidden in unstructured text. Modern NLP methods can be used to extract that information as structured data. In this section, we mainly focus on named entities and their relationships. These could be mentions of companies in news articles, credit reports, emails, or official filings. The extracted entities can be categorized and linked to a knowledge graph. Several of those are publicly accessible and cover a significant amount of relations, namely, Wikidata [77], the successor of DBpedia [34], and Freebase [9], as well as YAGO [76]. However, they are far from complete and usually general-purpose, so that specific domains or details might not be covered. Thus, it is essential to extend them automatically using company-internal documents or domain-specific texts.

The extraction of named entities is called *names entity recognition* (NER) [23] and comprises two steps: first, detecting the boundaries of the mention within the string of characters and second, classifying it into types such as ORGANIZATION, PERSON, or LOCATION. Through *named entity linking* (NEL) [70],[4] a mention is matched to its corresponding entry in the knowledge graph (if already known). An unambiguous assignment is crucial for integrating newly found information into a knowledge graph. For the scope of this chapter, we consider a fact to be a relation between entities. The most naïve approach is to use entity co-occurrence in text. *Relationship extraction* (RELEX) identifies actual connections stated in the text, either with an *open* or *closed* approach. In a closed approach, the relationships are restricted to a predefined set of relations, whereas the goal with an open approach is to extract all connections without restrictions.

Figure 1 shows a simplified example of a company network extracted from a small text excerpt. Instead of using the official legal names, quite different colloquial names, acronyms, or aliases are typically used when reporting about companies.

[4] Also called *entity resolution*, *entity disambiguation*, *entity matching*, or *record linkage*.

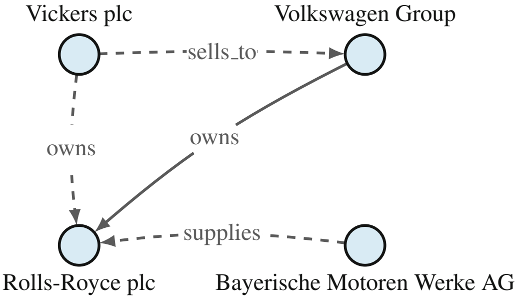

Fig. 1 Network of information extracted from the excerpt: *VW purchased Rolls-Royce & Bentley from Vickers on 28 July 1998. From July 1998 until December 2002, BMW continued to supply engines for the Rolls-Royce Silver Seraph (Excerpt from https://en.wikipedia.org/wiki/Volkswagen_Group. Accessed on 22.02.2020).*

There are three main challenges in entity linking: 1) *name variations* as shown in the example with "VW" and "Volkswagen"; 2) *entity ambiguity*, where a mention can also refer to multiple different knowledge graph entries; and 3) *unlinkable entities* in the case that there is no corresponding entry in the knowledge graph yet. The resulting graph in Fig. 1 depicts a sample knowledge graph generated from facts extracted from the given text excerpt. Besides the explicitly mentioned entities and relations, the excerpt also contains many implied relationships; for example, a sold company is owned by someone else after the sell. Further, relationships can change over time, leading to edges that are only valid for a particular time. This information can be stored in the knowledge graph and, e.g., represented through different types of edges in the graph. Through *knowledge graph completion*, it is possible to estimate the probability whether a specific relationship between entities exists [74].

In the remainder of this section, we provide a survey of techniques and tools for each of the three steps mentioned above: NER (Sect. 2.2), NEL (Sect. 2.2), and RELEX (Sect. 2.3).

2.1 Named Entity Recognition (NER)

The first step of the pipeline for knowledge graph construction from text is to identify mentions of named entities. Named entity recognition includes several subtasks, namely, identifying proper nouns and the boundaries of named entities and classifying the entity type. The first work in this area was published in 1991 and proposed an algorithm to automatically extract company names from financial news to build a database for querying [54, 46]. The task gained interest with MUC-6, a shared task to distinguish not only types, such as person, location, organization, but also numerical mentions, such as time, currency, and percentages [23]. Traditionally, research in this area is founded in computational linguistics, where the goal is to parse and describe the natural language with statistical rule-based methods. The foundation for that is to correctly tokenize the unstructured text, assign part-of-speech tags (also known as POS tagging), and create a parse tree that describes the sentence's dependencies and overall structure. Using this information, linguists defined rules that describe typical patterns for named entities.

Handcrafted rules were soon replaced by machine learning approaches that use tags mentioned above and so-called surface features. These surface features describe syntactic characteristics, such as the number of characters, capitalization, and other derived information. The most popular supervised learning methods for the task are hidden Markov models and conditional random fields due to their ability to derive probabilistic rules from sequence data [6, 42]. However, supervised learning requires large amounts of annotated training data. Bootstrapping methods can automatically label text data using a set of entity names as a seed. These semi-supervised methods do so by marking occurrences of these seed entities in the text and using contextual information to annotate more data automatically. For an overview of related work in that area, we refer to Nadeau et al. [47]. In recent years, deep learning approaches gained popularity. They have the advantage that they do not require sophisticated pre-processing, feature engineering, or potentially error-prone POS tagging and dependency parsing. Especially recurrent neural networks are well suited since they take entire sequences of tokens or characters into account. For an overview of currently researched deep learning models for NER, we refer readers to the extensive survey by Yadav and Bethard [80].

Although the task of automatically identifying company names in text, there is still a lot of research dedicated to named entity recognition. Due to their structural heterogeneity, recognizing company names is particularly difficult compared to person or location names. Examples of actual German company names show the complexity of the task. Not only are some of the names very long ("Simon Kucher & Partner Strategy & Marketing Consultants GmbH"), they interleave abstract names with common nouns, person names, locations, and legal forms, for example: "Loni GmbH," "Klaus Traeger," and "Clean-Star GmbH & Co Autowaschanlage Leipzig KG." Whereas in English almost all capitalized proper nouns refer to named entities, it is significantly harder to find entity mentions in other languages, for example, in German, where all common nouns are capitalized [17]. Loster et al. [65, 36] dedicate a series of papers to the recognition of financial entities in text. In particular, they focus on correctly determining the full extent of a mention by using tries, which are tree structures, to improve dictionary-based approaches [39].

The wealth of publications and the availability of open-source libraries reflect the importance and popularity of NER. The following overview shows the most successful projects used in research and industry alike.

GATE ANNIE The General Architecture for Text Engineering (GATE),[5] first released in 1995, is an extensive and mature open-source Java toolkit for many

[5] https://gate.ac.uk/.

aspects of natural language processing and information extraction tasks. ANNIE, A Nearly-New Information Extraction system, is the component for named entity extraction implementing a more traditional recognition model [15]. GATE provides all necessary tools to build a complete system for knowledge graph construction in combination with other components.

NLTK The Natural Language Toolkit (NLTK),[6] first released in 2001, is one of the most popular Python libraries for natural language processing [7]. It provides a wealth of easy-to-use API for all traditional text processing tasks and named entity recognition capabilities.

OpenNLP The Apache OpenNLP project[7] is a Java library for the most common text processing tasks and was first released in 2004 [27]. It provides implementations of a wide selection of machine learning-based NLP research designed to extend data pipelines built with Apache Flink or Spark.

CoreNLP The Stanford NLP research group released their first version of CoreNLP[8] in 2006 as a Java library with Python bindings [52]. It is actively developed and provides NLP tools ranging from traditional rule-based approaches to models from recently published state-of-the-art deep learning research.

spaCy Only recently in 2015, spaCy[9] was released as a Python/Cython library, focusing on providing high performance in terms of processing speed. It also includes language models for over 50 languages and a growing community of extensions. After an initial focus on processing speed, the library now includes high-quality language models based on recent advances in deep learning.

For a detailed comparison of the frameworks mentioned above, we refer readers to the recently published study by Schmitt et al. [68].

2.2 *Named Entity Linking (NEL)*

The problem of linking named entities is rooted in a wide range of research areas (Fig. 2). Through named entity linking, the strings discovered by NER are matched to entities in an existing knowledge graph or extend it. Wikidata is a prevalent knowledge graph for many use cases. Typically, there is no identical string match for an entity mention discovered in the text and the knowledge graph. Organizations are rarely referred to by their full legal name, but rather an acronym or colloquial variation of the full name. For example, VW could refer to Vorwerk, a manufacturer for household appliances, or Volkswagen, which is also known as Volkswagen Group

[6] http://nltk.org/.

[7] https://opennlp.apache.org.

[8] https://nlp.stanford.edu/software/.

[9] https://spacy.io/.

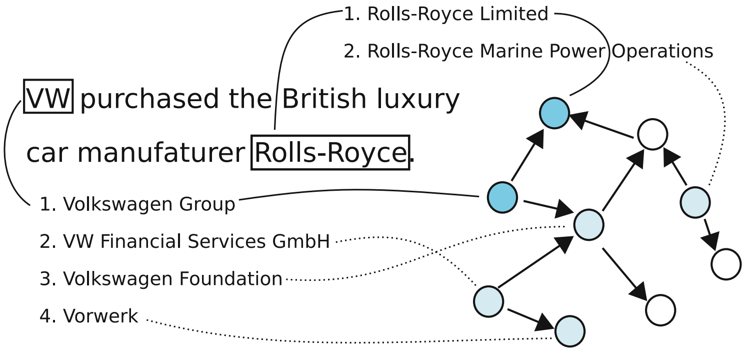
1. Rolls-Royce Limited
2. Rolls-Royce Marine Power Operations
VW purchased the British luxury
car manufaturer Rolls-Royce.
1. Volkswagen Group
2. VW Financial Services GmbH
3. Volkswagen Foundation
4. Vorwerk

Fig. 2 Example for ranking and linking company mentions to the correct entity in a set of candidates from the knowledge graph

or Volkswagen AG. At the time of writing, there are close to 80 entries in Wikidata[10] when searching for "Volkswagen," excluding translations, car models, and other non-organization entries. Entity linking approaches use various features to match the correct real-world entity. These features are typically based on the entity mention itself or information about the context in which it appeared. Thereby, they face similar challenges and use comparable approaches as research in record linkage and duplicate detection. Shen et al. [70] provide a comprehensive overview of applications, challenges, and a survey of the main approaches. As mentioned earlier, there are three main challenges when linking named entities, namely, name variations, entity ambiguity, and unlinkable entities. In this subsection, we discuss these challenges using examples to illustrate them better. We also present common solutions to resolve them and close with an overview of entity linking systems.

Name Variations A real-world entity is referred to in many different ways, such as the full official name, abbreviations, colloquial names, various known aliases, or simply with typos. These variations increase the complexity of finding the correct match in the knowledge base. For example, Dr. Ing. h.c. F. Porsche GmbH, Ferdinand Porsche AG, and Porsche A.G. are some name variations for the German car manufacturer Porsche commonly found in business news. Entity linking approaches traditionally take two main steps [70]. The first step selects candidate entries for the currently processed mention from the knowledge base. The second step performs the actual linking by choosing the correct candidate. The candidate generation reduces the number of possible matches, as the disambiguation can become computationally expensive. The most common approach is to use fuzzy string comparisons, such as an edit distance like the Levenshtein distance or the Jaccard index for overlapping tokens. Additionally, a few rules for name expansion can generate possible abbreviations or extract potential acronyms from names.

[10] https://www.wikidata.org/w/index.php?search=volkswagen.

These rules should use domain-specific characteristics, for example, common legal forms (Ltd. ↦ Limited) as well as names (International Business Machines ↦ IBM). If an existing knowledge base is available, a dictionary of known aliases can be derived.

Entity Ambiguity A mentioned entity could refer to multiple entries in the knowledge graph. For example, Volkswagen could not only refer to the group of car manufacturers but also the financial services, international branches, or the local car dealership. Only the context, the company mention appears in, may help identify the correct entry, by taking keywords within the sentence (local context) or the document (global context) into account. The entity disambiguation, also called *entity ranking*, selects the correct entry among the previously generated set of candidates of possible matches from the knowledge base. This second linking step aims to estimate the likelihood of a knowledge base entry being the correct disambiguation for a given mention. These scores create a ranking of candidates. Typically, the one with the highest score is usually chosen to be the correct match. Generally, ranking models follow either a supervised or unsupervised approach. Supervised methods that use annotated data mentions are explicitly linked to entries in the knowledge base to train classifiers, ranking models, probabilistic models, or graph-based methods. When there is no annotated corpus available, data-driven unsupervised learning or information retrieval methods can be used. Shen et al. [70] further categorize both approaches into three paradigms. *Independent ranking methods* consider entity mentions individually without leveraging relations between other mentions in the same document and only focusing on the text directly surrounding it. On the other hand, *collective ranking methods* assume topical coherence for all entity mentions in one document and link all of them collectively. Lastly, *collaborative ranking methods* leverage the textual context of similar entity mentions across multiple documents to extend the available context information.

Unlinkable Entities Novel entities have no corresponding entries in the knowledge graph yet. It is important to note that NEL approaches should identify such cases and not just pick the best possible match. Unlinkable entities may be added as new entries to the knowledge graph. However, this depends on the context and its purpose. Suppose $\mathsf{HBO_2}$ was found in a sentence and is supposed to be linked to a knowledge base of financial entities. If the sentence is about inorganic materials, this mention most likely refers to metaboric acid and should be dismissed, whereas in a pharmaceutical context, it might refer to the medical information systems firm HBO & Company. In that case, it should be added as a new entity and not linked to the already existing television network HBO. Entity linking systems deal with this in different ways. They commonly introduce a NIL entity, which represents a universal unlinkable entity, into the candidate set or a threshold for the likelihood score.

Other Challenges Growing size and heterogeneity of KGs are further challenges. Scalability and speed is a fundamental issue for almost all entity ranking systems. A key part to solve this challenge is a fast comparison function to generate candidates with a high recall to reduce the number of computations of similarity scores. State-of-the-art approaches that use vector representations have the advantage that nearest neighborhood searches within a vector space are almost constant [41]. However, training them requires large amounts of data, which might not be available in specific applications. Furthermore, targeted adaptations are not as trivial as with rule-based or feature-based systems. Another challenge for entity ranking systems are heterogeneous sources. Whereas multi-language requirements can be accounted for by separate models, evolving information over time imposes other difficulties. Business news or other sources continuously generate new facts that could enrich the knowledge graph further. However, with a growing knowledge graph, the characteristics of the data change. Models tuned on specific characteristics or trained on a previous state of the graph may need regular updates.

Approaches There are numerous approaches for named entity linking. Traditional approaches use textual fragments surrounding the entity mention to improve the linking quality over just using a fuzzy string match. Complex joint reasoning and ranking methods negatively influence the disambiguation performance in cases with large candidate sets. Zou et al. [83] use multiple bagged ranking classifiers to calculate a consensus decision. This way, they can operate on subsets of large candidate sets and exploit previous disambiguation decisions whenever possible. As mentioned before, not every entity mention can be linked to an entry in the knowledge graph. On the other hand, including the right entities in the candidate set is challenging due to name variations and ambiguities. Typically, there is a trade-off between the precision (also called linking correctness rate) of a system and its recall (also called linking coverage rate). For example, simply linking mentions of VW in news articles to the most popular entry in the knowledge graph is probably correct. All common aliases are well known and other companies with similar acronyms appear less frequently in the news, which leads to high precision and recall. In particular applications, this is more challenging. Financial filings often contain references to numerous subsidiaries with very similar names that need to be accurately linked. CohEEL is an efficient method that uses random walks to combine a precision-oriented and a recall-oriented classifier [25]. They achieve wide coverage while maintaining a high precision, which is of high importance for business analytics.

The research on entity linking shifted toward deep learning and embedding-based approaches in recent years. Generally, they learn high-dimensional vector representations of tokens in the text and knowledge graph entries. Zwicklbauer et al. [85] use such embeddings to calculate the similarity between an entity mention and its respective candidates from the knowledge graph. Given a set of training data in which the correct links are annotated in the text, they learn a robust similarity measure. Others use the annotated mentions in the training data as special tokens in the vocabulary and project words and entities into a common vector space [81, 21]. The core idea behind DeepType [53] is to support the linking process by providing type information about the entities from an existing knowledge graph to the disambiguation process, which they train in an end-to-end fashion. Such approaches require existing knowledge graphs and large sets of training data. Although this can be generated semi-automatically with open information extraction methods, maintaining a high quality can be challenging. Labeling high-quality training data manually is infeasible while maintaining high coverage. Active learning methods can significantly reduce the required amount of annotated data. DeepMatcher offers a ready-to-use implementation of a neural network that makes use of fully automatically learned attribute and word embeddings to train an entity similarity function with targeted human annotation [45].

2.3 Relationship Extraction (RELEX)

Relationship extraction identifies triples of two entities and their relation that appear in a text. Approaches follow one of two strategies: mining of *open-domain* triples or *fixed-domain* triples. In an open-domain setting, possible relations are not specified in advance and typically just use a keyword between two entities. Stanford's OpenIE [3] is a state-of-the-art information extraction system that splits sentences into sets of clauses. These are then shortened and segmented into triples. Figure 3 shows the relations extracted by OpenIE from the example used in Fig. 1. One such extracted triple would be (BMW, supply, Rolls-Royce).

Such a strategy is useful in cases where no training data or no ontology is available. An ontology is a schema (for a knowledge graph) that defines the types of possible relations and entities. In the following section, we provide more details on standardized ontologies and refinement. One disadvantage of open-domain extraction is that synonymous relationships lead to multiple edges in the knowledge graph. Algorithms can disambiguate the freely extracted relations after enriching the knowledge graph with data from all available text sources. In a fixed-domain setting, all possible relation types are known ahead of time. Defining a schema has the advantage that downstream applications can refer to predefined relation types. For example, in Fig. 1 we consider relations such as ORG owns ORG, which is implicitly matched by *"VW purchased Rolls-Royce."*

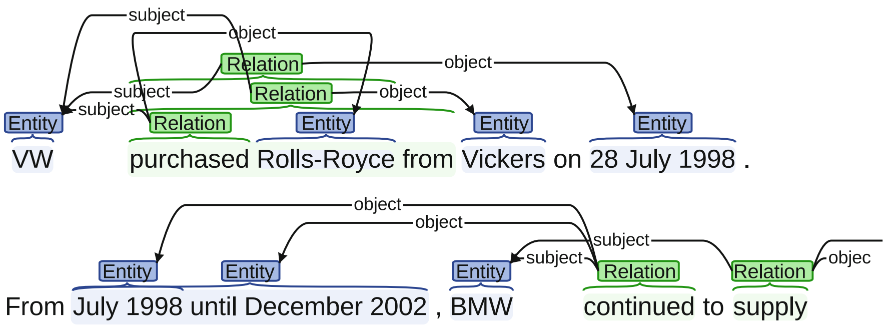

Fig. 3 Relations recognized by OpenIE in text from Fig. 1; output is visualized by CoreNLP (An online demo of CoreNLP is available at https://corenlp.run/.)

The naïve way to map relations mentioned in the text to a schema is to provide a dictionary for each relation type. An algorithm can automatically extend a dictionary from a few manually annotated sentences with relation triples or a seed dictionary. Agichtein and Gravano published the very famous *Snowball* algorithm, which follows this approach [1]. In multiple iterations, the algorithm grows the dictionary based on an initially small set of examples. This basic concept is applied in semi-supervised training to improve more advanced extraction models. The collection of seed examples can be expanded after every training iteration. This process is also called distant supervision. However, it can only detect relationship types already contained in the knowledge graph and cannot discover new relationship types. A comprehensive discussion of distant supervision techniques for relation extraction is provided by Smirnova [71]. Zuo et al. demonstrated the domain-specific challenges of extracting company relationships from text [84].

Recent approaches mostly focus on deep learning architectures to identify relations in a sequence of words. Wang et al. [78] use convolutional layers and attention mechanisms to identify the most relevant syntactic patterns for relation extraction. Others employ recurrent models to focus on text elements in sequences of variable length [33]. Early approaches commonly used conditional random fields (CRF) on parse trees, representing the grammatical structure and dependencies in a sentence. Nguyen et al. [48] combine modern neural BiLSTM architectures with CRFs for an end-to-end trained model to improve performance. Based on the assumption that if two entities are mentioned in the same text segment, Soares et al. [73] use BERT [16] to learn relationship embeddings. These embeddings are similar to dictionaries with the advantage that embedding vectors can be used to easily identify the matching relation type for ambiguous phrases in the text.

3 Refining the Knowledge Graph

In the previous section, we described the key steps in constructing a knowledge graph, namely, named entity extraction, entity linking, and relationship extraction. This process produces a set of triples from a given text corpus that forms a knowledge graph's nodes and edges. As we have shown in the previous section, compiling a duplicate-free knowledge graph is a complex and error-prone task. Thus, these triples need refinement and post-processing to ensure a high-quality knowledge graph. Any analysis based on this graph requires the contained information to be as accurate and complete as possible.

Manual refinement and standards are inevitable for high-quality results. For better interoperability, the Object Management Group, the standards consortium that defined UML and BPMN, among other things, specified the Financial Industry Business Ontology (FIBO).[11] This ontology contains standard identifiers for rela-

[11] https://www.omg.org/spec/EDMC-FIBO/BE/.

tionships and business entities. The Global Legal Identifier Foundation (GLEIF)[12] is an open resource that assigns unique identifiers to legal entities and contains statistics for around 1.5 million entries at the time of writing.

Using existing knowledge graphs as a reference together with standardized ontologies is a good foundation for the manual refinement process. However, the sheer size of these datasets requires support by automated mechanisms in an otherwise unattainable task. With CurEx, Loster et al. [37] demonstrate the entire pipeline of curating company networks extracted from text. They discuss the challenges of this system in the context of its application in a large financial institution [38]. Knowledge graphs about company relations are also handy beyond large-scale analyses of the general market situation. For example, changes in the network, as reported in SEC filings,[13] are of particular interest to analysts. Sorting through all mentioned relations is typically impractical. Thus, automatically identifying the most relevant reported business relationships in newly released filings can significantly support professionals in their work. Repke et al. [60] use the surrounding text, where a mentioned business relation appears, to create a ranking to enrich dynamic knowledge graphs. There are also other ways to supplement the available information about relations. For example, a company network with weighted edges can be constructed from stock market data [29]. The authors compare the correlation of normalized stock prices with relations extracted from business news in the same time frame and found that frequently co-mentioned companies oftentimes share similar patterns in the movements of their stock prices.

Another valuable resource for extending knowledge graphs are internal documents, as they contain specialized and proprietary domain knowledge. For example, the graph can also be extended beyond just company relations and include key personnel and semantic information. In the context of knowledge graph refinement, it is essential to provide high-quality and clean input data to the information extraction pipeline. The Enron Corpus [30], for example, has been the basis for a lot of research in many fields. This corpus contains over 500,000 emails from more than 150 Enron employees. The text's structure and characteristics in emails are typically significantly different from that of news, legal documents, or other reports. With Quagga,[14] we published a deep learning-based system to pre-process email text [55]. It identifies the parts of an email text that contains the actual content. It disregards additional elements, such as greetings, closing words, signatures, or automatically inserted meta-data when forwarding or replying to emails. This meta-data could extend the knowledge graph with information about who is talking to whom about what, which is relevant for internal investigations.

[12] https://search.gleif.org/.

[13] https://www.sec.gov/edgar.shtml.

[14] https://github.com/HPI-Information-Systems/QuaggaLib.

4 Analyzing the Knowledge Graph

Knowledge about the structure of the market is a highly valuable asset. This section focuses on specific applications in the domain of business intelligence for economics and finance. Especially financial institutions have to have a detailed overview of the entire financial market, particularly the network of organizations in which they invest. Therefore, Ronnqvist et al. [63] extracted bank networks from text to quantify interrelations, centrality, and determinants.

In Europe, banks are required by law to estimate their systemic risk. The network structure of the knowledge graph allows the investigation of many financial scenarios, such as the impact of corporate bankruptcy on other market participants within the network. In this particular scenario, the links between the individual market participants can be used to determine which companies are affected by bankruptcy and to what extent. Balance sheets and transactions alone would not suffice to calculate that risk globally, as it only provides an ego-network and thus a limited view of the market. Thus, financial institutions have to integrate their expertise in measuring the economic performance of their assets and a network of companies to simulate how the potential risk can propagate. Constantin et al. [14] use data from the financial network and market data covering daily stock prices of 171 listed European banks to predict bank distress.

News articles are a popular source of information for analyses of company relationships. Zheng and Schwenkler demonstrate that company networks extracted from news can be used to measure financial uncertainty and credit risk spreading from a distressed firm [82]. Others also found that the return of stocks reflects economic linkages derived from text [67]. We have shown that findings like this are controversial [29]. Due to the connectedness within industry sectors and the entire market, stock price correlation patterns are very common. Large companies and industry leaders heavily influence the market and appear more frequently in business news than their smaller competitors. Additionally, news are typically slower than movements on the stock market, as insiders receive information earlier through different channels. Thus, observation windows have to be in sync with the news cycle for analyses in this domain.

News and stock market data can then be used to show, for example, how the equity market volatility is influenced by newspapers [4]. Chahrour et al. [11] make similar observations and construct a model to show the relation between media coverage and demand-like fluctuations orthogonal to productivity within a sector. For models like this to work, company names have to be detected in the underlying texts and linked to the correct entity in a knowledge graph. Hoberg and Phillips extract an information network from product descriptions in 10-K statements filed with the SEC [26]. With this network, they examine how industry market structure and competitiveness change over time.

These examples show that knowledge graphs extracted from text can model existing hypotheses in economics. A well-curated knowledge graph that aggregates large amounts of data from a diverse set of sources would allow advanced analyses and market simulations.

5 Exploring the Knowledge Graph

Knowledge graphs of financial entities enable numerous downstream tasks. These include automated enterprise valuation, identifying the sentiment toward a particular company, or discovering political and company networks from textual data. However, knowledge graphs can also support the work of accountants, analysts, and investigators. They can query and explore relationships and structured knowledge quickly to gather the information they need. For example, visual analytics helps to monitor the financial stability in company networks [18]. Typically, such applications display small sub-graphs of the entire company network as a so-called node-link diagram. Circles or icons depict companies connected by straight lines. The most popular open-source tools for visualizing graphs are Cytoscape [49] and Gephi [5]. They mostly focus on visualization rather than capabilities to interactively explore the data. Commercial platforms, on the other hand, such as the NUIX Engine[15] or Linkurious,[16] offer more advanced capabilities. These include data pre-processing and analytics frequently used in forensic investigations, e.g., by journalists researching the Panama Papers leak.

There are different ways to visualize a network, most commonly by node-link diagrams as described above. However, already with a small number of nodes and edges, the readability is hard to maintain [51]. Edge bundling provides for better clarity of salient high-level structures [35]. The downside is that individual edges can become impossible to follow. Other methods for network visualization focus on the adjacency matrix of the graph. Each row and column corresponds to a node in the graph and cells are colored according to the edge weight or remain empty. The greatest challenge is to arrange the rows and columns in such a way that salient structures become visible. Sankey diagrams are useful to visualize hierarchically clustered networks to show the general flow of information or connections [12]. For more details have a look at the excellent survey of network visualization methods by Gibson et al. [22]. There is no one best type of visualization. It depends on the specific application to identify the ideal representation to explore the data.

Repke et al. developed “Beacon in the Dark” [59], a platform incorporating various modes of exploration. It includes a full system pipeline to process and integrate structured and unstructured data from email and document corpora. It also consists of an interface with coordinated multiple views to explore the data in a topic sphere (semantics), by tags that are automatically assigned, and the communication and entity network derived from meta-data and text. The system goes beyond traditional approaches by combining communication meta-data and integrating additional information using advanced text mining methods and social network analysis. The objectives are to provide a data-driven *overview* of the dataset to determine initial leads without knowing anything about the data. The system also

[15] NUIX Analytics extracts and indexes knowledge from unstructured data (https://www.nuix.com).

[16] Linkurious Enterprise is a graph visualization and analysis platform (https://linkurio.us/).

offers extensive filters and rankings of available information to focus on relevant aspects and finding necessary data. With each interaction, the interface components update to provide the appropriate context in which a particular information snippet appears.

6 Semantic Exploration Using Visualizations

Traditional interfaces for knowledge graphs typically only support node-to-node exploration with basic search and filtering capabilities. In the previous section, we have shown that some of them also integrate the underlying source data. However, they only utilize the meta-data, not the semantic context provided by the text in which entities appear. Furthermore, prior knowledge about the data is required to formulate the right queries as a starting point for exploration. In this section, we focus on methods to introduce semantics into the visualization of knowledge graphs. This integration enables users to explore the data more intuitively and provides better explanatory navigation.

Semantic information can be added based on the context in which entity mentions appear. The semantics could simply be represented by keywords from the text surrounding an entity. The benefit is that users not only interact with raw network data but with descriptive information. Text mining can be used to automatically enrich the knowledge about companies, for example, by assigning the respective industry or sentiment toward products or an organization itself. Such an approach is even possible without annotated data. Topic models assign distributions of topics to documents and learn which words belong to which topics. Early topic models, such as latent semantic indexing, do so by correlating semantically related terms from a collection of text documents [20]. These models iteratively update the distributions, which is computationally expensive for large sets of documents and long dictionaries. Latent semantic analysis, the foundation for most current topic models, uses co-occurrence of words [31]. Latent Dirichlet allocation jointly models topics as distributions over words and documents as distributions over topics [8]. This allows to summarize large document collections by means of topics.

Recently, text mining research has shifted toward embedding methods to project words or text segments into a high-dimensional vector space. The semantic similarity between words can be calculated based on distance measures between their vectors. Initial work in that area includes word2vec [44] and doc2vec [32]. More recent popular approaches such as BERT [16] better capture the essential parts of a text. Similar approaches can also embed graph structures. RDF2vec is an approach that generates sequences of connections in the graph leveraging local information about its sub-structures [62]. They show some applications that allow the calculation of similarities between nodes in the graph. There are also specific models to incorporate text and graph data to optimize the embedding space. Entity-centric models directly assign vectors to entities instead of just tokens. Traditionally, an embedding model assumes a fixed dictionary of known words or character ngrams that form these words. By annotating the text with named entity information before training the model, unique multi-word entries in the dictionary directly relate to known entities. Almasian et al. propose such a model for entity-annotated texts [2]. Other interesting approaches build networks of co-occurring words and entities. TopExNet uses temporal filtering to produce entity-centric networks for topic exploration in news streams [75]. For a survey of approaches and applications of knowledge graph embeddings, we refer the readers to [79].

Topic models, document embeddings, and entity embeddings are useful tools for systematic data analysis. However, on their own, they are not directly useable. In the context of book recommendations, embeddings have been used to find similar books using combinations of embeddings for time and place of the plot [61]. Similar approaches could be applied in the domain of financial entities, for example, to discover corresponding companies in a different country. In use cases without prior knowledge, it might be particularly helpful to get an overview of all the data. Also, for monitoring purposes, a bird's-eye view of the entire dataset can be beneficial. The most intuitive way is to organize the information in the form of an interactive map. Sarlin et al. [66] used self-organizing maps to arrange economic sectors and countries to create maps. Coloring the maps enables them to visually compare different financial stability metrics across multiple time frames around periods with high inflation rates or an economic crisis.

The idea of semantic landscapes is also popular in the area of patent research. The commercial software ThemeScape by Derwent[17] produces landscapes of patents that users can navigate similar to a geographical map. Along with other tools, they enable experts to find related patents or identify new opportunities quickly. Smith et al. built a system to transform token co-occurrence information in texts to semantic patterns. Using statistical algorithms, they generate maps of words that can be used for content analysis in knowledge discovery tasks [72]. Inspired by that, the New York Public Library made a map of parts of their catalog.[18] Therefore, they use a force-based network layout algorithm to position the information. It uses the analogy of forces that attract nodes to one another when connected through an edge or otherwise repel them. The network they use is derived from co-occurring subject headings and terms, which were manually assigned tags to organize their catalog. Sen et al. created a map of parts of the Wikipedia in their Cartograph project [69]. This map, as shown in Fig. 4, uses embedded pages about companies and dimensionality reduction to project the information on a two-dimensional canvas [40, 50]. Structured meta-data about pages is used to compute borders between "countries" representing different industry sectors. Maps like this provide an intuitive alternative interface for users to discover related companies. Most recently, the Open Syllabus Project[19] released their interactive explorer. Like Cartograph, this enables users to navigate through parts of the six million syllabi

[17] https://clarivate.com/derwent.

[18] https://www.nypl.org/blog/2014/07/31/networked-catalog.

[19] Open Syllabus Explorer visualization shows the 164,720 texts (http://galaxy.opensyllabus.org/).

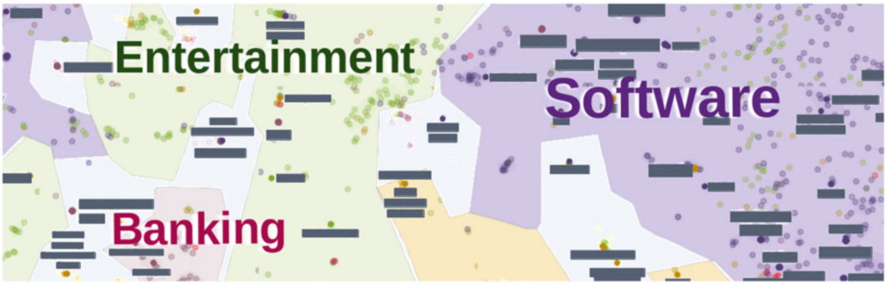

Fig. 4 Screenshot of part of the Cartograph map of organizations and their sectors

collected by the project. To do so, they first create a citation network of all publications contained in the visualization. Using this network, they learn a node embedding [24] and reduce the number of dimensions for rendering [43].

The approaches presented above offer promising applications in business analytics and exploring semantically infused company networks. However, even though the algorithms use networks to some extent, they effectively only visualize text and rely on manually tagged data. Wikipedia, library catalogs, and the syllabi corpus are datasets that are developed over many years by many contributors who organize the information into structured ontologies. In business applications, data might not always have this additional information available, and it is too labor-intensive to curate the data manually. Furthermore, when it comes to analyzing company networks extracted from text, the data is comprised of both the company network and data provenance information. The methods presented above only visualize either the content data or the graph structure. In data exploration scenarios, the goal of getting a full overview of the dataset at hand is insurmountable with current tools. We provide a solution that incorporates both, the text sources *and* the entity network, into exploratory landscapes [56]. We first embed the text data and then use multiple objectives to optimize for a good network layout and semantically correct the layout of source documents during the dimensionality reduction [58]. Figure 5 shows a small demonstration of the resulting semantic-infused network

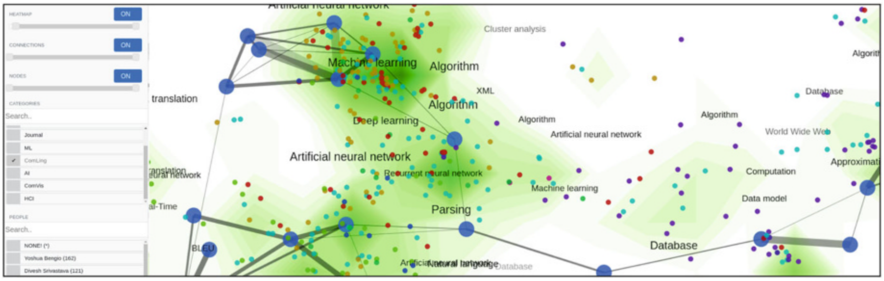
HEATMAP ON
CONNECTIONS ON
NODES ON
CATEGORIES
Search..
Journal
ML
☑ CompLing
AI
CompVis
HCI
PEOPLE
Search..
NONE! (*)
Yoshua Bengio (162)
Divesh Srivastava (121)

Fig. 5 Screenshot of the *MODiR* interface prototype showing an excerpt of a citation network

layout [57]. Users exploring such data, e.g., journalists investigating leaked data or young scientists starting research in an unfamiliar field, need to be able to interact with the visualization. Our prototype allows users to explore the generated landscape as a digital map with zooming and panning. The user can select from categories or entities to shift the focus, highlight characterizing keywords, and adjust a heatmap based on the density of points to only consider related documents. We extract region-specific keywords and place them on top of the landscape. This way, the meaning of an area becomes clear and supports fast navigation.

7 Conclusion

In this chapter, we provided an overview of methods to automatically construct a knowledge graph from text, particularly a network of financial entities. We described the pipeline starting from named entity recognition, over linking and matching those entities to real-world entities, to extracting the relationships between them from text. We emphasized the need to curate the extracted information, which typically contains errors that could negatively impact its usability in subsequent applications. There are numerous use cases that require knowledge graphs connecting economic, financial, and business-related information. We have shown how these knowledge graphs are constructed from heterogeneous textual documents and how they can be explored and visualized to support investigations, analyses, and decision making.

References

1. Agichtein, E., & Gravano, L. (2000). *Snowball*: Extracting relations from large plain-text collections. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)* (pp. 85–94). New York, NY, USA: ACM Press.
2. Almasian, S., Spitz, A., & Gertz, M. (2019). Word embeddings for entity-annotated texts. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. Lecture Notes in Computer Science (vol. 11437, pp. 307–322). Berlin: Springer.
3. Angeli, G., Premkumar, M. J. J., & Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 344–354). Stroudsburg, PA, USA: Association for Computational Linguistics.
4. Baker, S. R., Bloom, N., Davis, S. J., & Kost, K. J. (2019). *Policy news and stock market volatility*. Working Paper 25720, National Bureau of Economic Research.
5. Bastian, M., Heymann, S., Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the International Semantic Web Conference (ISWC)*. Palo Alto, CA, USA: The AAAI Press.
6. Bikel, D. M., Miller, S., Schwartz, R. M., & Weischedel, R. M. (1997). Nymble: A high-performance learning name-finder. In *Applied Natural Language Processing Conference (ANLP)* (pp. 194–201). Stroudsburg, PA, USA: Association for Computational Linguistics.
7. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA, USA: O'Reilly.

8. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research **3***(Jan), 993–1022.
9. Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM Conference on Management of Data (SIGMOD)* (pp. 1247–1250).
10. Chabin, M. A. (2017). Panama papers: A case study for records management? *Brazilian Journal of Information Science: Research Trends, 11*(4), 10–13.
11. Chahrour, R., Nimark, K., & Pitschner, S. (2019). *Sectoral media focus and aggregate fluctuations*. Swedish House of Finance Research Paper Series 19–12, SSRN.
12. Chang, C., Bach, B., Dwyer, T., & Marriott, K. (2017). Evaluating perceptually complementary views for network exploration tasks. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI)* (pp. 1397–1407). New York, Ny, USA: ACM Press.
13. Coddington, M. (2015). Clarifying journalism's quantitative turn: A typology for evaluating data journalism, computational journalism, and computer-assisted reporting. *Digital Journalism, 3*(3), 331–348.
14. Constantin, A., Peltonen, T. A., & Sarlin, P. (2018). Network linkages to predict bank distress. *Journal of Financial Stability, 35*, 226–241.
15. Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 168–175). Stroudsburg, PA, USA: Association for Computational Linguistics.
16. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)* (pp. 4171–4186). Stroudsburg, PA, USA: Association for Computational Linguistics.
17. Faruqui, M., & Padó, S. (2010). Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of the Conference on Natural Language Processing (KONVENS)* (pp. 129–133).
18. Flood, M. D., Lemieux, V. L., Varga, M., & Wong, B. W. (2016). The application of visual analytics to financial stability monitoring. *Journal of Financial Stability, 27*, 180–197.
19. Franke, K., & Srihari, S.N. (2007). Computational forensics: Towards hybrid-intelligent crime investigation. In *Proceedings of the International Symposium on Information Assurance and Security (IAS)* (pp. 383–386). New York City, NY, USA: IEEE.
20. Furnas, G. W., Deerwester, S. C., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., et al. (1988). Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the ACM Conference on Information Retrieval (SIGIR)* (pp. 465–480). New York City, NY, USA: ACM Press.
21. Ganea, O., & Hofmann, T. (2017). Deep joint entity disambiguation with local neural attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 2619–2629). Stroudsburg, PA, USA: Association for Computational Linguistics.
22. Gibson, H., Faith, J., & Vickers, P. (2013). A survey of two-dimensional graph layout techniques for information visualisation. *Information Visualization, 12*(3–4), 324–357.
23. Grishman, R., & Sundheim, B. (1996). Message understanding conference- 6: A brief history. In *Proceedings of the International Conference on Computational Linguistics (COLING)* (pp. 466–471).
24. Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 855–864). New York City, NY, USA: ACM Press.
25. Grütze, T., Kasneci, G., Zuo, Z., & Naumann, F. (2016). CohEEL: Coherent and efficient named entity linking through random walks. *Journal of Web Semantics, 37–38*, 75–89.
26. Hoberg, G., & Phillips, G. (2016). Text-based network industries and endogenous product differentiation. *Journal of Political Economy, 124*(5), 1423–1465.

27. Ingersoll, G., Morton, T., & Farris, A. (2012). *Taming text*. Shelter Island, NY, USA: Manning Publications.
28. Karthik, M., Marikkannan, M., & Kannan, A. (2008). An intelligent system for semantic information retrieval information from textual web documents. In *International Workshop on Computational Forensics (IWCF)* (pp. 135–146). Heidelberg: Springer.
29. Kellermeier, T., Repke, T., & Krestel, R. (2019). Mining business relationships from stocks and news. In V. Bitetta, I. Bordino, A. Ferretti, F. Gullo, S. Pascolutti, & G. Ponti (Eds.), *Proceedings of MIDAS 2019*, Lecture Notes in Computer Science (vol. 11985, pp. 70–84). Heidelberg: Springer.
30. Klimt, B., & Yang, Y. (2004). The Enron corpus: A new dataset for email classification research. In *Proceedings of the European Conference on Machine Learning (ECML)* (pp. 217–226). Heidelberg: Springer.
31. Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*(2–3), 259–284.
32. Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 1188–1196). Brookline, Ma, USA: JMLR Inc. and Microtome Publishing.
33. Lee, J., Seo, S., & Choi, Y. S. (2019). Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing. *Symmetry, 11*(6), 785.
34. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., et al. (2015). DBpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web, 6*(2), 167–195.
35. Lhuillier, A., Hurter, C., & Telea, A. (2017). State of the art in edge and trail bundling techniques. *Computer Graphics Forum, 36*(3), 619–645.
36. Loster, M., Hegner, M., Naumann, F., & Leser, U. (2018). Dissecting company names using sequence labeling. In *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen" (LWDA). CEUR Workshop Proceedings* (vol. 2191, pp. 227–238). CEUR-WS.org.
37. Loster, M., Naumann, F., Ehmueller, J., & Feldmann, B. (2018). Curex: A system for extracting, curating, and exploring domain-specific knowledge graphs from text. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)* (pp. 1883–1886). New York City, NY, USA: ACM Press.
38. Loster, M., Repke, T., Krestel, R., Naumann, F., Ehmueller, J., Feldmann, B., et al. (2018). The challenges of creating, maintaining and exploring graphs of financial entities. In *Proceedings of the International Workshop on Data Science for Macro-Modeling with Financial and Economic Datasets (DSMM@SIGMOD)* (pp. 6:1–6:2). New York City, NY, USA: ACM Press.
39. Loster, M., Zuo, Z., Naumann, F., Maspfuhl, O., & Thomas, D. (2017). Improving company recognition from unstructured text by using dictionaries. In *Proceedings of the International Conference on Extending Database Technology (EDBT)* (pp. 610–619). OpenProceedings.org.
40. Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR), 9*, 2579–2605.
41. Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(4), Art. No. 8594636, 824-836.
42. McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)* (pp. 188–191). Stroudsburg, PA, USA: Association for Computational Linguistics.
43. McInnes, L., & Healy, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *CoRR, abs/1802.03426*.
44. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* (pp. 3111–3119). San Diego, CA, USA: NIPS Foundation.

45. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., et al. (2018). Deep learning for entity matching: A design space exploration. In *Proceedings of the ACM Conference on Management of Data (SIGMOD)* (pp. 19–34). New York City, NY, USA: ACM Press.
46. Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3–26.
47. Nadeau, D., Turney, P. D., & Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In L. Lamontagne & M. Marchand (Eds.), *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*. Lecture Notes in Computer Science (vol. 4013, pp. 266–277). Berlin: Springer.
48. Nguyen, D. Q., & Verspoor, K. (2019). End-to-end neural relation extraction using deep biaffine attention. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. Lecture Notes in Computer Science (vol. 11437, pp. 729–738). Berlin: Springer.
49. Otasek, D., Morris, J. H., Bouças, J., Pico, A. R., & Demchak, B. (2019). Cytoscape automation: Empowering workflow-based network analysis. *Genome Biology, 20*(1), 1–15.
50. Pezzotti, N., Lelieveldt, B. P., van der Maaten, L., Höllt, T., Eisemann, E., & Vilanova, A. (2017). Approximated and user steerable t-SNE for progressive visual analytics. *IEEE Transactions on Visualization and Computer Graphics (TVCG), 23*(7), 1739–1752.
51. Pohl, M., Schmitt, M., & Diehl, S. (2009). Comparing the readability of graph layouts using eyetracking and task-oriented analysis. In *Computational Aesthetics 2009: Eurographics Workshop on Computational Aesthetics*, Victoria, British Columbia, Canada, 2009 (pp. 49–56).
52. Qi, P., Dozat, T., Zhang, Y., & Manning, C. D. (2018). Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 160–170). Stroudsburg, PA, USA: Association for Computational Linguistics.
53. Raiman, J., & Raiman, O. (2018). DeepType: Multilingual entity linking by neural type system evolution. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 5406–5413). Palo Alto, CA, USA: AAAI Press.
54. Rau, L. F. (1991). Extracting company names from text. In *Proceedings of the IEEE Conference on Artificial Intelligence Application* (vol. 1, pp. 29–32). Piscataway: IEEE.
55. Repke, T., & Krestel, R. (2018). Bringing back structure to free text email conversations with recurrent neural networks. In *Proceedings of the European Conference on Information Retrieval (ECIR)* (pp. 114–126). Heidelberg: Springer.
56. Repke, T., & Krestel, R. (2018). Topic-aware network visualisation to explore large email corpora. In *International Workshop on Big Data Visual Exploration and Analytics (BigVis), Proceedings of the International Conference on Extending Database Technology (EDBT)* (pp. 104–107). CEUR-WS.org.
57. Repke, T., & Krestel, R. (2020). Exploration interface for jointly visualised text and graph data. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)* (pp. 73–74). Geneva: ACM Press.
58. Repke, T., & Krestel, R. (2020). Visualising large document collections by jointly modeling text and network structure. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)* (pp. 279–288). Geneva: ACM Press.
59. Repke, T., Krestel, R., Edding, J., Hartmann, M., Hering, J., Kipping, D., et al. (2018). Beacon in the dark: A system for interactive exploration of large email corpora. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)* (pp. 1871–1874). New York, NY, USA: ACM Press.
60. Repke, T., Loster, M., & Krestel, R. (2017). Comparing features for ranking relationships between financial entities based on text. In *Proceedings of the International Workshop on Data Science for Macro-Modeling with Financial and Economic Datasets (DSMM@SIGMOD)* (pp. 12:1–12:2). New York, NY, USA: ACM Press.
61. Risch, J., Garda, S., & Krestel, R. (2018). Book recommendation beyond the usual suspects - embedding book plots together with place and time information. In *Proceedings of the International Conference on Asia-Pacific Digital Libraries (ICADL)*. Lecture Notes in Computer Science (vol. 11279, pp. 227–239). Berlin: Springer.

62. Ristoski, P., Rosati, J., Noia, T. D., Leone, R. D., & Paulheim, H. (2019). RDF2Vec: RDF graph embeddings and their applications. *Semantic Web, 10*(4), 721–752.
63. Rönnqvist, S., & Sarlin, P. (2015). Bank networks from text: interrelations, centrality and determinants. *Quantitative Finance, 15*(10), 1619–1635.
64. Ruder, S., Vulic, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research (JAIR), 65*, 569–631.
65. Samiei, A., Koumarelas, I., Loster, M., & Naumann, F. (2016). Combination of rule-based and textual similarity approaches to match financial entities. In *Proceedings of the International Workshop on Data Science for Macro-Modeling, (DSMM@SIGMOD)* (pp. 4:1–4:2). New York, NY, USA: ACM Press.
66. Sarlin, P. (2013). Exploiting the self-organizing financial stability map. *Engineering Applications of Artificial Intelligence, 26*(5–6), 1532–1539.
67. Scherbina, A., & Schlusche, B. (2015). *Economic linkages inferred from news stories and the predictability of stock returns*. AEI Economics Working Papers 873600, American Enterprise Institute. https://ideas.repec.org/p/aei/rpaper/873600.html
68. Schmitt, X., Kubler, S., Robert, J., Papadakis, M., & Traon, Y.L. (2019). A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate. In *International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 338–343). Piscataway: IEEE.
69. Sen, S., Swoap, A. B., Li, Q., Boatman, B., Dippenaar, I., Gold, R., et al. (2017). Cartograph: Unlocking spatial visualization through semantic enhancement. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)* (pp. 179–190). Geneva: ACM Press.
70. Shen, W., Wang, J., & Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering, 27*(2), 443–460.
71. Smirnova, A., & Cudré-Mauroux, P. (2019). Relation extraction using distant supervision: A survey. *ACM Computing Surveys, 51*(5), 106:1–106:35.
72. Smith, A. E., & Humphreys, M. S. (2006). Evaluation of unsupervised semantic mapping of natural language with leximancer concept mapping. *Behavior Research Methods, 38*(2), 262–279.
73. Soares, L. B., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 2895–2905). Stroudsburg, PA, USA: Association for Computational Linguistics.
74. Socher, R., Chen, D., Manning, C. D., & Ng, A. Y. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* (pp. 926–934).
75. Spitz, A., Almasian, S., & Gertz, M. (2019). TopExNet: Entity-centric network topic exploration in news streams. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)* (pp. 798–801). New York, NY, USA: ACM Press.
76. Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). YAGO: A core of semantic knowledge. In *Proceedings of the International World Wide Web Conference (WWW)* (pp. 697–706).
77. Vrandečić, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Communications of the ACM, 57*(10), 78–85.
78. Wang, L., Cao, Z., de Melo, G., & Liu, Z. (2016). Relation classification via multi-level attention CNNs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 1298–1307). New York, NY, USA: ACM Press.
79. Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering, 29*(12), 2724–2743.
80. Yadav, V., & Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the International Conference on Computational Linguistics (COLING)* (pp. 2145–2158). Stroudsburg, PA, USA: Association for Computational Linguistics.

81. Yamada, I., Shindo, H., Takeda, H., & Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)* (pp. 250–259). Stroudsburg, PA, USA: Association for Computational Linguistics.
82. Zheng, H., & Schwenkler, G. (2020). *The network of firms implied by the news*. ESRB Working Paper Series 108, European Systemic Risk Board.
83. Zuo, Z., Kasneci, G., Grütze, T., & Naumann, F. (2014). BEL: Bagging for entity linking. In J. Hajic & J. Tsujii (Eds.), *Proceedings of the International Conference on Computational Linguistics (COLING)* (pp. 2075–2086). Stroudsburg, PA, USA: Association for Computational Linguistics.
84. Zuo, Z., Loster, M., Krestel, R., & Naumann, F. (2017). Uncovering business relationships: Context-sensitive relationship extraction for difficult relationship types. In *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings, CEUR Workshop Proceedings* (vol. 1917, p. 271). CEUR-WS.org.
85. Zwicklbauer, S., Seifert, C., & Granitzer, M. (2016). Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the ACM Conference on Information Retrieval (SIGIR)* (pp. 425–434). New York, NY, USA: ACM Press.