Towards Knowledge Graph Construction from Entity Co-occurrence

Nicolas Heist[0000−0002−4354−9138]

Data and Web Science Group, University of Mannheim, Germany
nico@informatik.uni-mannheim.de

Abstract. The construction of knowledge graphs from resources on the Web is a topic that gained a lot of attention in recent years, especially with the uprising of large-scale cross-domain knowledge graphs like DBpedia and YAGO. Their successful exploitation of Wikipedia's structural elements like infoboxes and categories gives way to the thought that there is still a huge potential for approaches focused on structural elements of web documents. In this work, we present our research idea towards further exploitation of semi-structured data with a focus on entity co-occurrence. We want to explore the potential of co-occurrence patterns in varying contexts and test their generality when applying them to the Document Web. An overview of the state of the art is given and we show how our three-phased approach for the extraction of co-occurrence patterns fits in. Two planned experiments for the construction of knowledge graphs based on Wikipedia and the Document Web are sketched. Finally, potentials and limitations of the approach are discussed.

Keywords: Knowledge Acquisition · Knowledge Graph Construction · Entity Co-occurrence · Information Extraction · Pattern Extraction.

1 Problem Statement

The Web is a vast source of structured and unstructured data. Unfortunately, extracting knowledge from this pool of data is a non-trivial task. In the recent years, however, extraordinary progress has been made in extracting knowledge from the Web and persisting it in a machine-readable format. Google coined the term "Knowledge Graph" (KG) in 2012 to describe such stores of knowledge that contain ontological information describing a certain domain as well as facts describing the state of the world with respect to the domain.[1] Many application areas like question answering [10], entity disambiguation [15], and text categorization [9] as well as concrete applications like search engines and AI assistants profit heavily from the availability of domain-specific as well as cross-domain KGs.

Large-scale cross-domain KGs like DBpedia [14] and YAGO [22] contain millions of entities and several hundred millions of facts. Both of them rely on

[1] https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

Wikipedia as their prime source of knowledge (with YAGO additionally using WordNet [17] to build its ontology). Another commonality is their exploitation of Wikipedia's structural properties to extract high-quality information. While DBpedia depends on infoboxes to gain information about articles, YAGO exploits the category system to build a taxonomy of articles and discover relationships between them. This is, of course, no coincidence, because extracting information from (semi-)structured sources yields results that are yet unmatched by approaches working on unstructured data only (YAGO reports accuracy values of about 95% [22]). Being aware that only a small part of the information on the web is available in a (semi-)structured format, we believe that there is still huge potential in exploiting the structuredness of data with the aim of constructing large-scale KGs.

In this paper, we describe our research idea for the construction of a KG using entity co-occurrence. In principle, we want to discover patterns in web documents that indicate relationships between the included entities. Instead of focusing on relationships between two entities at a time, however, the aim is to identify patterns including groups of entities that are related twofold: they are connected on the document surface (e.g. they appear in the same table column) and have a semantic connection (e.g. they are both persons who live in Berlin).

We define the problem as follows: $r_{sur}(e_1, e_2)$ denotes a relation between entities e_1 and e_2 that manifests in the surface of the underlying text corpus (i.e. they are somehow connected on the document level) and $r_{sem}(e_1, e_2)$ describes a semantic relationship between e_1 and e_2 (i.e. they have an arbitrary common property). We denote the document corpus with D, entities in a document $d \in D$ with E_d, and entities used to extract a co-occurrence pattern p with E_p. A pattern p is uniquely defined by specific relations r_{sur} and r_{sem}, and the source document d. For every document $d \in D$ we want to find a set of patterns P_d with

$$P_d = \{p_{d,r_{sur},r_{sem}} | \forall e_1, e_2 \in E_p : e_1, e_2 \in E_d \wedge r_{sur}(e_1, e_2) \wedge r_{sem}(e_1, e_2)\}$$

To be able to extract information from arbitrary documents, the extracted document-level sets are fused into a document-independent set P.

Based on our problem definition, we pose the following research questions:
RQ1: Is it possible to discover arbitrary entity co-occurrence patterns locally (i.e. within a bounded context like Wikipedia) as well as globally (on the Web)?
RQ2: Can co-occurrence patterns be grouped into different types of patterns and if so, how do these groups differ in their performance?
RQ3: How well can (groups of) co-occurrence patterns be generalized so that they can be applied to arbitrary web documents?

The remainder of this paper is organized as follows: In the following section we describe the current state of the art. In Section 3 we elaborate on our research idea based on two specific examples. Section 4 describes the research methodology in detail and in Section 5 we sketch our planned experiments. Finally, we conclude with a discussion of the general research idea in Section 6.

2 State of the Art

A large amount of publications tackle the problem of KG construction [23]. [5] identify four groups of approaches that can be characterized by their choice of data sources and ontology: (1) approaches that exploit the structuredness of Wikipedia and either use a predefined ontology (like DBpedia [14]) or extract their ontology from the underlying structured data (like YAGO [22]); (2) open information extraction approaches that work without an ontology and extract information from the whole web (e.g. [6]); (3) approaches that use a fixed ontology and also target the whole web (e.g. KnowledgeVault [5], NELL [3]); and finally (4) approaches that target the whole web, but construct taxonomies (is-a hierarchies) instead (e.g. [12, 24]).

While inspired by approaches from (1), our research idea can best be categorized into group (3) as the aim is to extract knowledge from the whole web and use an existing ontology. Consequently, we will focus on approaches from those groups in the remainder of this section. Besides DBpedia [14] and YAGO [22], two more Wikipedia-based approaches are WiBi [7] and DBTax [8]. While the authors of WiBi generate a taxonomy by iteratively extracting hypernymy-relationships from Wikipedia's article and category network, the authors of DBTax use an unsupervised approach to scan the category tree of Wikipedia for prominent nodes which for themselves already form a complete taxonomy. Inspired by an analysis about list pages in Wikipedia from Paulheim and Ponzetto [20], [13] strive to augment DBpedia by exploiting the fact that entities in Wikipedia's list pages are all instances of a common concept. They use statistical methods to discover the common type of a list page in order to assign it to the entities which are lacking it. In [19] the authors use Wikipedia's tables to extract multiple millions of facts. By bootstraping their approach with data from DBpedia, they are able to apply machine learning in order to extract facts with a precision of about 81.5%. [11] exploit the structuredness of abstracts of Wikipedia pages to extract facts related to the subject of the page and mentioned entities. They use a supervised classification approach using only features that are language-independent like the position of the mentioned entity in the abstract or the types of the mentioned entity.

The never-ending language learner NELL [3] cyclically crawls a corpus of one billion web pages to continuously learn new facts by harvesting text patterns. KnowledgeVault [5] gathers facts from web documents by extracting them from text, HTML tables, the DOM tree, and *schema.org* annotations. To verify their validity, they compare the extracted facts with existing knowledge in Freebase [1]. Most of their extractors are designed to discover relations between two entities (e.g. for the extraction from the DOM tree the lexicalized path between two entities is used as feature vector). Only for the extraction of facts from HTML tables they consider groups of entities as the relations in tables are usually expressed between whole columns. Various other approaches use HTML tables (e.g. [21]) or structured markup (e.g. [25]) for the extraction of facts. Nevertheless, none of these define a generic approach for the extraction of facts between multiple entities using arbitrary structures of a web document.

3 Approach

Figure 1 shows two exemplary settings for an application of the approach. With background knowledge from an existing KG, it can be applied to any corpus of web documents. Figure 1a displays a Wikipedia list page of persons who are all related on the surface level (i.e. they all appear in the same enumeration) and on the semantic level (i.e. they are all fictional journalists). As Wikipedia entities are closely linked to DBpedia, the entities referenced in these lists can be linked to their respective counterpart in DBpedia automatically, thus making it easy to automatically find semantic commonalities between them. By using the information about the entities, we can identify groups of entities and extract a pattern for the recognition of such entities on a Wikipedia list page. In this case such a pattern could identify the first entity of every enumeration point as a fictional journalist.

Figure 1b shows a more generic setting where the extraction of the pattern is rather difficult as entities on this page are not linked already and a navigation bar is not as standardized as an enumeration. Nevertheless, there are various entity recognition and linking tools that can help with the former problem (e.g. [15]). Regarding the latter problem it is worth noting that there is a steady increase in adoption of Open Source software [4] and especially Web Content Management Systems, thus making it likely to find more and more websites with standardized components.

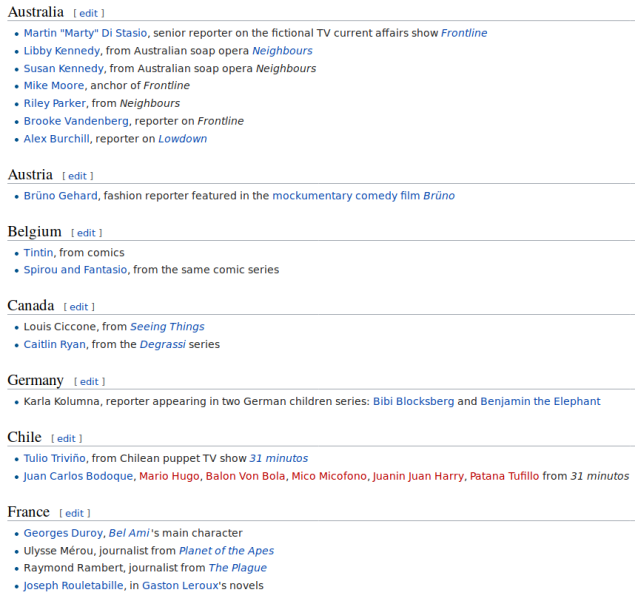
In both scenarios the whole underlying document corpus can be scanned for semantically related entities within specific documents in order to discover their relation on the surface of the document. Fusing and generalizing the extracted patterns can then yield in (a) a pattern for the extraction of persons from Wikipedia list pages and (b) a pattern for the extraction of persons from navigation bars. When additional contextual information is included in the pattern (e.g. information about the domain) it may even be possible to define general patterns for the extraction of more specific types like journalists or scientists, respectively.

4 Methodology

4.1 Knowledge Graph Construction

The foundation of our processing pipeline form a KG that is used as seed (KG_s) and a corpus of web documents D. The extraction itself can be separated into three phases: Pattern Extraction, Pattern Fusion and Pattern Application.

Pattern Extraction: If necessary, entities in D are located and linked to KG_s. Applying distant supervision [18] and the local closed world assumption [5] to KG_s and D, we can gather data for the extraction of patterns. Specficially, we want to find patterns comprising entities that are related through specific relations r_{sur} and r_{sem}. Paths in the DOM tree can serve as feature vectors for arbitrary web documents but depending on the corpus D it might make sense to use more specific feature vectors (like Wiki markup when using Wikipedia as



Australia [edit]
- Martin "Marty" Di Stasio, senior reporter on the fictional TV current affairs show *Frontline*
- Libby Kennedy, from Australian soap opera *Neighbours*
- Susan Kennedy, from Australian soap opera *Neighbours*
- Mike Moore, anchor of *Frontline*
- Riley Parker, from *Neighbours*
- Brooke Vandenberg, reporter on *Frontline*
- Alex Burchill, reporter on *Lowdown*

Austria [edit]
- Brüno Gehard, fashion reporter featured in the mockumentary comedy film *Brüno*

Belgium [edit]
- Tintin, from comics
- Spirou and Fantasio, from the same comic series

Canada [edit]
- Louis Ciccone, from *Seeing Things*
- Caitlin Ryan, from the *Degrassi* series

Germany [edit]
- Karla Kolumna, reporter appearing in two German children series: Bibi Blocksberg and Benjamin the Elephant

Chile [edit]
- Tulio Triviño, from Chilean puppet TV show *31 minutos*
- Juan Carlos Bodoque, Mario Hugo, Balon Von Bola, Mico Micofono, Juanin Juan Harry, Patana Tufillo from *31 minutos*

France [edit]
- Georges Duroy, *Bel Ami*'s main character
- Ulysse Mérou, journalist from *Planet of the Apes*
- Raymond Rambert, journalist from *The Plague*
- Joseph Rouletabille, in Gaston Leroux's novels

(a) Excerpt from Wikipedia's *List of fictional journalists*

People
- Intro
- Professors
- Administration
- Researchers
  - Dr. Sanja Stajner
  - Dr. Ioana Hulpus
  - Dr. Melisachew Wudage Chekol
  - Dr. Christian Meilicke
  - Dr. Federico Nanni
  - Dr. Dmitry Ustalov
  - Taha Alhersh
  - Alexander Diete
  - Manuel Fink
  - Nicolas Heist
  - Sven Hertling
  - Jakob Huber
  - Amirhossein Kardoost
  - Elena Kuss
  - Anne Lauscher
  - Oliver Lehmberg

(b) Navigation bar for researchers of the DWS group

(a) https://en.wikipedia.org/wiki/List_of_fictional_journalists
(b) https://dws.informatik.uni-mannheim.de/en/people/researchers

Fig. 1. Examples of semi-structured data sources of the Document Web

data corpus). The output of this phase is a (possibly empty) set of patterns P_d for every $d \in D$.

Pattern Fusion: Two patterns p_1 and p_2 may be merged together if their respective relations r_{sur} and r_{sem} are equal, subsume one another, or can both be subsumed by a more general relation. For r_{sur} a subsumption can mean that only the part of the DOM tree that p_1 and p_2 have in common is used as pattern. Regarding r_{sem}, a pattern that indentifies scientists and a pattern that identifies journalists are merged into a pattern that identifies persons. Patterns can then be ranked by their accuracy on the extraction data and their support in D in order to filter out irrelevant patterns. As an output we have a set P with generalized patterns for D.

Pattern Application: As a last step, the patterns in P can either be applied to D or, depending on the generality of the patterns, to any other corpus of web documents to extract new entities and facts. Note that while new entities can be extracted with any co-occurrence pattern, facts are always extracted in the context of the relation r_{sem} of the respective pattern.

Finally, an approach similar to the one applied in [3] could be used to transform this sequential extraction pipeline into an iterative one. After an initial pattern extraction, the discovered surface relations are applied to D in order to find new semantic relations, and vice versa.

4.2 Knowledge Graph Evaluation

We plan to evaluate our resulting KGs on three levels: To get a first impression of the results, we evaluate intrinsically by comparing metrics like size or coverage with other KGs of the domain (see [26] for a comprehensive overview of KG metrics). An extrinsic evaluation is conducted with the help of human judges in order to get an absolute estimation of the quality of our results. Due to the tremendous size of large-scale KGs, we plan to use crowd-sourced evaluation tools like Amazon Mechanical Turk.[2] Finally, we will perform a task-based evaluation by analyzing whether the performance of applications increase when our KG is used instead of others.

5 Planned Experiments

For an exploration of the potential of co-occurrence patterns, our first prototype will be implemented in a constrained environment where r_{sur} is restricted to a specific type of patterns. Using DBpedia as seed KG and Wikipedia as document corpus, the idea is to construct a KG from the category tree and connected list pages. While the category tree already served for many publications as the backbone of a taxonomy, list pages have only been used in few occasions (see Section 2). This may be due to the fact that list pages, unlike categories, have no precisely defined structure and their hierarchical organization within Wikipedia is rather implicit. In general, a list page is a Wikipedia page with a title starting with *List of*. [13] have analyzed 2,000 list pages and identified three common layouts: They appear either as enumeration, table or in an arbitrary unstructured format, while the latter appears less frequent. Hence, we will focus on the former two types due to their structured nature and frequency. Co-occurrence patterns can then be derived as explained in Section 3. The English version of DBpedia contains 212,175 list pages in its latest release[3], so we are positive that a lot of still hidden knowledge can be extracted with this approach.

Using the insights gained from our first prototype, we will subsequently perform an experiment in an unconstrained environment using the whole web as document corpus. Here, we strive to extract patterns where r_{sur} and r_{sem} can have arbitrary forms. The Common Crawl[4] will serve as our source of documents. Instead of linking entities in the crawl on our own, we plan to use semantic annotations on web pages (e.g. using Microdata or RDFa format [16]). Consequently, the pipeline described in Section 4 will be applicable. The most recent Web Data Commons crawl for semantic annotations[5] contains almost 40 billion triples in several million domains and various approaches (cf. [5, 25]) have successfully utilized them for their experiments. Hence, we see this as a promising setup for the large-scale extraction of co-occurrence patterns.

[2] https://www.mturk.com/
[3] Pages starting with *List of* in http://downloads.dbpedia.org/2016-10/core-i18n/en/labels_en.ttl.bz2
[4] http://commoncrawl.org/
[5] http://webdatacommons.org/structureddata/#results-2017-1

6 Discussion

Our approach for the construction of KGs from entity co-occurrence is designed to exploit arbitrary document structures that contain related entities. It thus extends the state of the art as existing approaches either focus on relations between two entities ([3, 5]) or treat only special cases of document structures like tables ([19, 21]).

The approach bears potential as it works orthogonally to the existing approaches by focusing on harvesting patterns formed by multiple entities. Consequently, it might be possible to extract information that is yet untouched since, as soon as a co-occurrence pattern is found, no evidence for a certain fact in the immediate surroundings of an entity is necessary in order to extract it. The main limitation of our approach is the inability to extend the ontology of the seed KG. Depending on the richness of the ontology, some relations might not be representable, resulting in a potential loss of information. Furthermore, it is yet unexplored how efficiently co-occurrence patterns can be extracted (on large scale) and whether it is necessary to include additional contextual information into the patterns in order to create document-independent ones.

7 Acknowledgements

I would like to thank Heiko Paulheim for his guidance and support in the realization of this work and Stefan Dietze for his elaborate review und valuable suggestions for the general direction of my research.

References

1. Bollacker, K., Evans, C., et al.: Freebase: a collaboratively created graph database for structuring human knowledge. In: 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. AcM (2008)
2. Brewster, C., Alani, H., et al.: Data driven ontology evaluation (2004)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI. vol. 5, p. 3. Atlanta (2010)
4. D Macredie, R., Mijinyawa, K.: A theory-grounded framework of open source software adoption in smes. European Journal of Information Systems **20**(2), 237–250 (2011)
5. Dong, X., Gabrilovich, E., et al.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 601–610. ACM (2014)
6. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Conference on empirical methods in natural language processing. pp. 1535–1545 (2011)
7. Flati, T., Vannella, D., et al.: Two is bigger (and better) than one: the wikipedia bitaxonomy project. In: 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 945–955 (2014)

8. Fossati, M., Kontokostas, D., Lehmann, J.: Unsupervised learning of an extensive and usable taxonomy for dbpedia. In: 11th International Conference on Semantic Systems. pp. 177–184. ACM (2015)
9. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. Journal of Artificial Intelligence Research **34**, 443–498 (2009)
10. Harabagiu, S.M., Moldovan, D.I., et al.: Falcon: Boosting knowledge for answer engines. In: TREC. vol. 9, pp. 479–488 (2000)
11. Heist, N., Paulheim, H.: Language-agnostic relation extraction from wikipedia abstracts. In: International Semantic Web Conference. pp. 383–399. Springer (2017)
12. Hertling, S., Paulheim, H.: Webisalod: providing hypernymy relations extracted from the web as linked open data. In: International Semantic Web Conference. pp. 111–119. Springer (2017)
13. Kuhn, P., Mischkewitz, S., Ring, N., Windheuser, F.: Type inference on wikipedia list pages. Informatik 2016 (2016)
14. Lehmann, J., Isele, R., Jakob, M., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
15. Mendes, P.N., Jakob, M., et al.: Dbpedia spotlight: shedding light on the web of documents. In: 7th international conference on semantic systems. pp. 1–8. ACM (2011)
16. Meusel, R., Petrovski, P., Bizer, C.: The webdatacommons microdata, rdfa and microformat dataset series. In: International Semantic Web Conference. pp. 277–292. Springer (2014)
17. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
18. Mintz, M., Bills, S., et al.: Distant supervision for relation extraction without labeled data. In: Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. pp. 1003–1011 (2009)
19. Muñoz, E., Hogan, A., Mileo, A.: Using linked data to mine rdf from wikipedia's tables. In: 7th ACM international conference on Web search and data mining. pp. 533–542. ACM (2014)
20. Paulheim, H., Ponzetto, S.P.: Extending dbpedia with wikipedia list pages. NLP-DBPEDIA@ ISWC **13** (2013)
21. Ritze, D., Lehmberg, O., et al.: Profiling the potential of web tables for augmenting cross-domain knowledge bases. In: 25th international conference on world wide web. pp. 251–261 (2016)
22. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
23. Weikum, G., Theobald, M.: From information to knowledge: harvesting entities and relationships from web sources. In: 29th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 65–76. ACM (2010)
24. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: A probabilistic taxonomy for text understanding. In: 2012 ACM SIGMOD International Conference on Management of Data. pp. 481–492. ACM (2012)
25. Yu, R., Gadiraju, U., et al.: Knowmore-knowledge base augmentation with structured web markup. Semantic Web Journal, IOS Press (2017)
26. Zaveri, A., Rula, A., Maurino, A., et al.: Quality assessment for linked data: A survey. Semantic Web **7**(1), 63–93 (2016)