Entity-Relationship Extraction from Wikipedia Unstructured Text

Radityo Eko Prasojo

KRDB Research Centre, Free University of Bozen Bolzano, BZ 39100, Italy
rprasojo@unibz.it

Abstract. Wikipedia has been the primary source of information for many automatically-generated Semantic Web data sources. However, they suffer from incompleteness since they largely do not cover information contained in the unstructured texts of Wikipedia. Our goal is to extract structured entity-relationships in RDF from such unstructured texts, ultimately using them to enrich existing data sources. Our extraction technique is aimed to be topic-independent, leveraging grammatical dependency of sentences and semantic refinement. Preliminary evaluations of the proposed approach have shown some promising results.

Keywords: Relation extraction, knowledge base generation, Wikipedia

1 Problem Statement

The Semantic Web envisions a web of data that can be easily processed by machines. In contrast, only a small portion of information available on the Web is in a machine readable format. For this reason, Semantic Web data sources suffer from incompleteness since they do not cover unstructured information, which represent the major part of the Web. A typical example are knowledge bases such as YAGO [27] and DBpedia [4], extracted from Wikipedia. These knowledge bases exploit infoboxes of Wikipedia articles. Hence, they can answer questions like the birth date or the political party of Barack Obama. However, they are unable to give further information like Barack Obama's favorite sport team because it is present only in the unstructured part of the Wikipedia article. This problem calls for the development of effective strategies that transform Web unstructured content into a machine-readable format. Consequently, we can cover more information and facilitate automatic data processing.

In this work, we consider Wikipedia as our primary source of information because of its coverage and cleanliness [12]. Being the most popular Internet encyclopedia, Wikipedia hosts articles related to any kind of topics. As mentioned earlier, each article typically also has an infobox, which contains a selected set of information related to the article. Most RDF data sources nowadays contain information that is extracted only from the infobox, where an accurate extraction is guaranteed thanks to its semi-structured nature. Our goal is to go further and extract RDF triples from the unstructured part of Wikipedia articles. For example, in the Wikipedia article of Barack Obama[1] it is mentioned that "Obama is a supporter of the Chicago White Sox". This information can be represented as the following triple: `BarackObama supporterOf ChicagoWhiteSox`. Such a

[1] https://en.wikipedia.org/wiki/Barack_Obama

triple can then be added to any existing data source, making it more complete. The extraction of such entity-relationship information from all Wikipedia articles into RDF triples has an ultimate goal of enriching existing data sources.

2 Relevancy

In the Semantic Web, introducing more high-quality RDF data in efficient manner is an important issue. Many online applications, such as search engines, social medias, or news websites, have started to utilize information from the Semantic Web. For example, a news article talking about Barack Obama can be enriched by some information that is previously known outside the news. If this information is stored in RDF, as opposed to in human language, it can be retrieved and processed more quickly and effectively.

3 Challenges and Related Work

The main challenge in this work is that there are two different problems that need to be simultaneously dealt with. The first is the relation extraction (RE) problem, that is, given a text containing entities, we syntactically extract relations between them. The second is the knowledge representation (KR) problem, that is, the extracted relations should always follow a well-defined schema, semantics, or *ontology*. This is an important issue because we want to combine all the extracted relations to enrich existing data sources. Without handling the knowledge representation properly, one can just extract all possible relations without considering whether, for example, two relations are equivalent and should be merged together, or whether additional semantic details can be mined from the sentence in order to correctly represent a more complex fact.

To illustrate the challenge, consider the previous example about Obama's favorite sports team. In the same article, it is also mentioned that "in his childhood and adolescence was a fan of the Pittsburgh Steelers". From the sentence, let us extract the following triple: `BarackObama fanOf PittsburghSteelers`. From the extraction point of view, the result is already correct. However, from the KR point of view it is not good enough. First, because the predicate `supporterOf` and `fanOf` are equivalent in this context and should be merged. Second, since Obama *was* a fan of Pittsburgh Steelers *in his childhood*, it is suggested that the fact happened in the past and therefore adding time information in the representation is necessary to differentiate it with the first example. Now, the next challenge is how we represent a complex fact. One solution is that we append time information into the predicate, resulting in `wasFanOf`, and then we specify the sub-predicate relation between `wasFanOf` and `fanOf`. Another solution is that we keep using `fanOf`, and then we leverage RDF reification to append time information as a separate triple. There may be other possible solutions, and deciding which one is the best is most of the times difficult.

Because of the above challenges, extracting relations from Wikipedia unstructured text has been overlooked for data-source generation purposes. Automatically-generated data sources like YAGO [27] and DBpedia [4] extract relations only from infoboxes which provide two advantages. First, they are semi-structured ensuring accurate extraction. Second, they provide schema for ontol-

ogy building, ensuring a semantically well-defined data source. In YAGO case, the ontology is enhanced by exploiting Wikipedia category pages and WordNet.

On the other hand, previous work that focused on relation extraction from unstructured text did not concern much with the representation issue because they do not have the goal of building or enriching Semantic Web data sources. Typically, they use some pre-defined schema that is taken from infoboxes. As such, they showed that they can find relations between entities only if they are also present in infoboxes. Various technique has been used to achieve this goal, including grammatical dependency exploitation [13] and anaphoric coreferences [20]. Some other work relied on existing data sources to find relations in the text [6] [11] [17] [21]. Because of this restriction, they cannot discover new relations that are not previously known by the infoboxes or the data sources. Beyond Wikipedia domain, there are also efforts with similar objectives [3] [7] [19] [26] [29].

Some work tried to deal with the knowledge representation issue to some extent. Yan et al. [30] tried to detect equivalent relations by leveraging surface patterns. However, they did not deal with complex fact representation. The Never Ending Language Learning (NELL) [18] is an ongoing project that aims to build a knowledge base to understand human language on the Web. Part of it is by storing relations between entities, which is done not only over Wikipedia but also other websites. However, until now a final result in the form of a structured RDF data sources containing entity relationship has not been finished yet. Similarly, FRED [10] is machine reader that transforms a free natural language text into RDF format, but it does not aim to create a well-constructed KB as a result. On the other hand, Google Knowledge Vault [5] succeeded to automatically build a KB from free text. However, they rely on distant supervision from pre-existing KB so they still cannot find new relations. Because representing relations of topic-independent articles is difficult, some other work focused only on a specific type of articles, like occupation of people [9] or events [14] [22]. Some other approaches focused on a specific type of relations instead, for example taxonomic [2], hyponymy [28], or numerical relations [15].

From knowledge representation literature (untied to information extraction), we consider works on expressing temporal information [24], epistemic modality [1], and complex fact in general [23] to be relevant.

4 Research questions

Summarizing the challenges explained in Section 3, we define our research questions as follows:

1. How do we extract relations between entities from the unstructured text of Wikipedia articles?
2. How should we represent the relations? More specifically, how should we represent complex facts?
3. How should we deal with the extraction problem and representation problem? Is there a way to structure the two problems in a good way?

In the next section, we explain our hypothesis and proposed approaches to answer the above research questions.

5 Hypothesis and Proposed Approaches

To syntactically extract relations, we plan to leverage grammatical dependency parsing of sentences. We hypothesize that it should be effective to detect any kind of relations between entities in a text because a relation is always in the form of subject, predicate, and object (i.e., a triple). Figure 1 shows an example of a grammatically annotated sentence using StanfordNLP [16]. In this example, the aim is to extract the triple `Obama supporterOf ChicagoWhiteSox`. To do this, first we look at the entity occurrences, which are Barack Obama and Chicago White Sox.[2] Then, we check how the relation can be extracted by looking at the dependency. We observe that both entities are connected via an `nsubj` dependency and an `nmod:of` dependency which share a head at the word "supporter". By leveraging these two dependencies, we can correctly extract the relation.

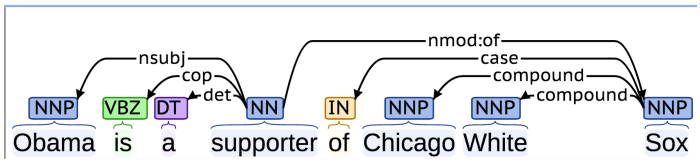

Fig. 1. Grammatical dependency example

We then observe that this extraction can be applied to a more general case, forming an extraction rule r_1 which goes as follows: for any sentence that contains a form of $[s \xleftarrow{\texttt{nsubj}} h \xrightarrow{\texttt{nmod:}prep} o]$ where s is an entity subject, *prep* is some preposition, o is an entity object, and h is the common head words, then we can extract from the sentence a relation $< s\ \texttt{concat}(h, prep)\ o >$ where `concat` is the string concatenation function. Relations that can be extracted by this rule, for example, can include the predicates `livedIn`, `marriedTo`, etc. We hypothesize that this extraction rule should be effective, that is, it can extract simple relations from simple sentences with high precision. We identify simple relations by looking at the fact that the relation can be represented using only one triple and the predicate can be represented correctly by simple concatenation of words, while simple sentences simply mean that they have a simple grammatical dependency structure, as defined by the extraction rule.

We further make two hypotheses. First, there should exist simple extraction rules other than r_1 which leverage other kinds of grammatical dependency. We define R as the set of all such simple extraction rules. Second, there should exist other kinds of sentences such that if we apply only the grammatical parsing, then the resulted relation is not good enough because some details are missing, and that a more sophisticated representation of the relation is needed. Recall the example in Section 3 about Pittsburgh Steelers being Obama's favorite football club during his childhood. In order to understand that "his childhood" refers to a time in the past, a syntactical parsing is not enough. A semantic refinement is necessary. From our two hypotheses, we now understand that there are two factors that determine the complexity of the problem. First, the complexity

[2] In Wikipedia, these entity annotations are typically given. In the case of missing annotations, we will do a preprocessing first before extracting the relations.

of syntactical extraction which is based on the complexity of the grammatical dependency. Second, the complexity of knowledge representation, which is based on the necessity of representing complex facts. Based on this observation, we define four difficulty classes of the problem, shown in Figure 2.

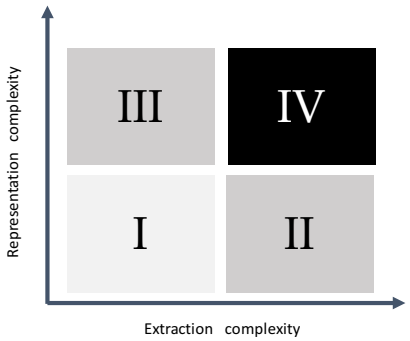

Fig. 2. The four difficulty classes of relation extraction

Class I contains sentences from which every extraction rule in R can be applied, which means that every correct relations can be extracted using simple grammatical extraction and simple representation. Class II contains sentences that may require a more complex grammatical dependency parsing that is not contained in R, but still have simple representation. Figure 3 shows an example of a sentence in this class. One can observe that the grammatical dependency is much more complicated than the one shown in Figure 1.

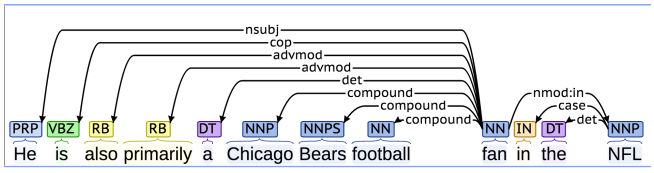

Fig. 3. Dependency parser example of Class II sentence

Class III and IV both contain sentences from which the extracted relations require a complex representation. For sentences in Class III, an extraction rule from R can still be applied but the result would need a further refinement. Figure 4 shows the previous example where the relation `Obama fanOf PittsburghSteeler` can be extracted using extraction rule r_1, but it is not precise because it does not include time information. On the other hand, a sentence in Class IV would require a possibly complex extraction rule that is not present in R.

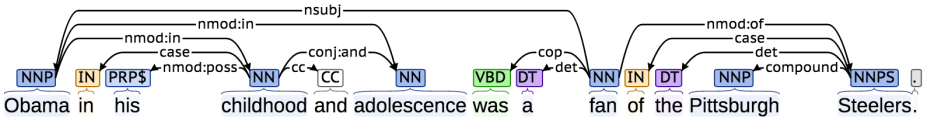

Fig. 4. Dependency parser example of Class III Sentence

We plan to develop our extraction approach by going through the difficulty classes starting from the simplest one. First, we will handle Class I sentences.

This is done by adding more rules to the set R. Each rule should have a high precision, which is set by a certain threshold. An extraction rule that does not perform above the threshold should not be included into R. Then we proceed with Class III since it requires simple extraction as Class I. we apply semantic refinement to find the missing details. We would investigate on possible ways of doing this, one of which is to use lexical database like WordNet [8]. At a later stage we would address both Class II and Class IV sentences since they require more sophisticated extraction. We assume that most sentences would be in either Class I or Class III, therefore leaving out the other two cases would keep the precision high without penalizing the recall. Finally, we will do another round of semantic refinement after we extract relations from all sentences, primarily to detect relations that should be merged.

6 Evaluation Plan

We plan to do two kinds of evaluation. The first one is to evaluate the performance of our extraction technique over a manually constructed ground-truth. We would separately evaluate our syntactical extraction and semantic refinement extraction in order to correctly assess the strengths and weaknesses of our approach. We also plan to compare our approach to existing related work as baselines.

The second one is to evaluate how good our extraction technique in finding new relations that are not previously present in a data source. We will use YAGO and DBpedia as the data sources for evaluation. To do this, we need to do mappings between our extracted relations and the respective data source. For this purpose, we plan to leverage available schemas such as schema.org.[3]

7 Preliminary Results

We have evaluated a part of our proposed approach over a small dataset containing 25 Wikipedia articles about famous people. Each article was first preprocessed by cleaning the noisy Wikipedia annotations and completing missing entity annotations using entity resolution techniques [25]. Then, we applied the extraction rules for Class I sentences, only. We included four extraction rules into R, one of them is the r_1. The other three rules are as follows: (1) one that handles passive sentences similar to r_1 using `nsubjpass` dependency, (2) one that handles direct object relation using `dobj` dependency and (3) one that also handles object relation by using `xcomp` dependency instead. By detecting multiple entity occurrences in a single sentence, we observed that from a total 9646 sentences, 4259 contain relations between entities. Among them, 1048 (24.6%) fall into Class I category. This is a quite significant amount, given that we can still add more rules into R. So our first observation is in line with our assumption that most sentences would be in either Class I or Class III.

We have also evaluated the precision and recall of our Class I extraction over the "Personal Life", "Early Life", "Life", and "Legacy" sections from each article. We were able to extract in total 205 relations. The result is shown in Table

[3] http://schema.org/

1. The precision and recall are shown in two ways: the first is the normal precision and recall, where the extraction is done solely by leveraging grammatical dependency parser. On the other hand, the ***SR*** precision and recall shows the performance of hypothetically doing a 100% accurate **S**emantic **R**efinement on top of the grammatical parsing result. One can observe that we have a promising early result in terms of precision, especially if later we can come up with an effective semantic refinement. However, we still need to improve the recall. We observed two main problems. First, our preprocessing should be improved, as there is still noise that hinder our extraction process. Second, we need to add more rules into R, as the four rules that we currently have are not enough.

Table 1. Preliminary result of extraction from Class I sentences

Prec	***Recall***	***Prec (SR)***	***Recall (SR)***
0.688	0.259	0.907	0.342

8 Reflections

We believe that this research work will lead to fruitful results. We have structured our work based on the four difficulty classes, which enable us to focus on the simplest thing at first, then later to extend it to more difficult cases. Also, our preliminary experiments have shown some promising results.

Acknowledgments

The author would like to thank Mouna Kacimi and Werner Nutt for their support and guidance as supervisors. The author would also like to thank Markus Zanker, Fariz Darari, and Simon Razniewski for their feedback.

References

1. L. Alonso-Ovalle and P. Menéndez-Benito. *Epistemic Indefinites: Exploring Modality Beyond the Verbal Domain*. Oxford University Press, USA, 2015.
2. P. Arnold and E. Rahm. Automatic extraction of semantic relations from wikipedia. *International Journal on Artificial Intelligence Tools*, 24(2), 2015.
3. M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.
4. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics*, 7(3):154–165, 2009.
5. X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th SIGKDD*, pages 601–610. ACM, 2014.
6. P. Exner and P. Nugues. Entity extraction: From unstructured text to DBpedia RDF triples. In *WoLE 2012*, pages 58–69. CEUR-WS, 2012.
7. A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the EMNLP*, pages 1535–1545. ACL, 2011.
8. C. Fellbaum. *WordNet*. Wiley Online Library, 1998.
9. D. Firas, L. Simon, and P. Nugues. Extraction of career profiles from wikipedia. In *Proceedings of the 1st Conference on Biographical Data in a Digital World*, 2015.
10. A. Gangemi, V. Presutti, D. R. Recupero, A. G. Nuzzolese, F. Draicchio, and M. Mongiovì. Semantic web machine reading with FRED. *SemWeb Journal*, 2016.
11. Y. Gu, W. Liu, and J. Song. Relation extraction from wikipedia leveraging intrinsic patterns. In *2015 IEEE/WIC/ACM WI-IAT*, volume 1, pages 181–186, Dec 2015.

12. B. Han, P. Cook, and T. Baldwin. Lexical normalization for social media text. *ACM Trans. Intell. Syst. Technol.*, 4(1):5:1–5:27, Feb. 2013.
13. A. Herbelot and A. Copestake. Acquiring ontological relationships from wikipedia using rmrs. In *In ISWC 2006 Workshop on Web Content*, 2006.
14. E. Kuzey and G. Weikum. Evin: building a knowledge base of events. In *Proceedings of the 23rd WWW Conference*, pages 103–106. WWW Steering Committee, 2014.
15. A. Madaan, A. Mittal, G. Ramakrishnan, and S. Sarawagi. Numerical relation extraction with minimal supervision. In *Proceedings of the 30th AAAI*, 2016.
16. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.
17. M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the JC of the 47th ACL and the 4th IJCNLP: Vol. 2-Vol. 2*, pages 1003–1011. ACL, 2009.
18. T. M. Mitchell, W. W. Cohen, E. R. H. Jr., P. P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. T. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *Proceedings of the 29th AAAI*, pages 2302–2310, 2015.
19. N. Nakashole, G. Weikum, and F. Suchanek. PATTY: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 JC on EMNLP and CoNLL*, pages 1135–1145. Association for Computational Linguistics, 2012.
20. D. P. Nguyen, Y. Matsuo, and M. Ishizuka. Relation extraction from wikipedia using subtree mining. In *Proceedings of the AAAI*, page 1414. MIT Press, 2007.
21. T.-V. T. Nguyen and A. Moschitti. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th ACL: Human Language Technologies: short papers-Volume 2*, pages 277–282. ACL, 2011.
22. M. Norrby and P. Nugues. Extraction of lethal events from wikipedia and a semantic repository. In *workshop on Semantic resources and semantic annotation for NLP and the Digital Humanities at NODALIDA 2015*, 2015.
23. N. Noy and A. Rector. Defining n-ary relations on the semantic web. Technical report, World Wide Web Consortium, 04 2006.
24. M. J. O'Connor and A. K. Das. *A Method for Representing and Querying Temporal Information in OWL*. Springer Berlin Heidelberg, 2011.
25. R. E. Prasojo, M. Kacimi, and W. Nutt. Entity and aspect extraction for organizing news comments. In *Proceedings of the 24th CIKM*, pages 233–242. ACM, 2015.
26. M. Schmitz, R. Bart, S. Soderland, O. Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 JC on EMNLP and CoNLL*, pages 523–534. ACL, 2012.
27. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, 6(3):203–217, 2008.
28. B. Wei, J. Liu, J. Ma, Q. Zheng, W. Zhang, and B. Feng. Motif-based hyponym relation extraction from wikipedia hyperlinks. *Knowledge and Data Engineering, IEEE Transactions on*, 26(10):2507–2519, 2014.
29. Y. Xu, M.-Y. Kim, K. Quinn, R. Goebel, and D. Barbosa. Open information extraction with tree kernels. In *HLT-NAACL*, pages 868–877, 2013.
30. Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, and M. Ishizuka. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the JC of the 47th ACL and the 4th IJCNLP: Vol. 2-Vol. 2*, pages 1021–1029. ACL, 2009.