Improving Open Relation Extraction via Sentence Re-Structuring

Jordan Schmidek **Denilson Barbosa**
Department of Computing Science
University of Alberta
Edmonton, AB, Canada
schmidek@ualberta.ca, denilson@ualberta.ca

Abstract

Information Extraction is an important task in Natural Language Processing, consisting of finding a structured representation for the information expressed in natural language text. Two key steps in information extraction are identifying the entities mentioned in the text, and the *relations* among those entities. In the context of Information Extraction for the World Wide Web, unsupervised relation extraction methods, also called *Open Relation Extraction* (ORE) systems, have become prevalent, due to their effectiveness without domain-specific training data. In general, these systems exploit part-of-speech tags or semantic information from the sentences to determine whether or not a relation exists, and if so, its predicate. This paper discusses some of the issues that arise when even moderately complex sentences are fed into ORE systems. A process for re-structuring such sentences is discussed and evaluated. The proposed approach replaces complex sentences by several others that, together, convey the same meaning and are more amenable to extraction by current ORE systems. The results of an experimental evaluation show that this approach succeeds in reducing the processing time and increasing the accuracy of the state-of-the-art ORE systems.

Keywords: Information Extraction, Open Relation Extraction, Sentence Re-Structuring

1. Introduction

The proliferation of massive text corpora on the Web has made Information Extraction (IE), often also called Text Analytics (Jurafsky and Martin, 2008), a very important task in Natural Language Processing in recent decades. The ultimate goal of an IE system is to extract the information in the text and represent it in a structured way (e.g., as a table or a graph), amenable to storage, indexing, and query processing by a standard database management system, or processing by a statistical analysis tool, among other applications. Despite their many differences, all IE systems perform two basic operations: *Named Entity Recognition* (NER), and *relation extraction*.

NER systems label (primarily) noun phrases as entities in the text. Relation extraction systems analyze the text between two named entities, and determine the relation between them (if one exists). The first relation extraction systems were supervised and specific to a single relation. In order to tackle massive text corpora containing a large number of unknown relations, Banko and Etzioni (2008), among others, proposed systems that tackle the Open Relation Extraction (ORE) problem: finding relations in text without *a priori* knowledge of which relations actually exist in the text. ORE Systems quickly became prevalent in Web information extraction applications where supervised methods do not scale.

A key ingredient of ORE systems is to rely on the output of NLP tools instead of the actual words in the sentence. Their fundamental goal is to exploit the relatively few grammatical constructs that express relations between entities. For example, many relations are expressed using a single verb between two entities as in "E_1 *married* E_2" or "E_1 *acquired* E_2". Similarly, other relations are expressed using a noun and a preposition, following a copula, as in "E_1 is the *CEO* of E_2" or "E_1 was *born* in E_2".

Clearly, the effectiveness of ORE systems depends on the sophistication and accuracy of the NLP tools they employ. Mesquita et al. (2013) characterize the state-of-the-art in ORE in these terms: Broadly speaking, existing ORE approaches can be grouped according to the level of sophistication of the NLP techniques they rely upon: (1) shallow parsing, (2) dependency parsing and (3) semantic role labelling. There is a clear cost-benefit trade-off established by this scale: the most efficient methods use "shallow" NLP techniques (e.g., POS tagging only) while the more effective ones are based on "deeper" NLP (e.g., dependency parsing). That paper also describes a new system, called EXEMPLAR, that borrows elements from all three groups, resulting in a system that is as accurate as the "deep" NLP methods but much faster.

> "The pattern is pretty much the same across the nation, said [Kris Knapton]#PER, a spokesman for [Metra]#ORG, a commuter rail service in northeastern [Illinois]#LOC."

Figure 1: Example complex sentence with relations. Named entities are identified and tagged with a type (PER for person, ORG for organization and LOC for location).

This paper considers an orthogonal issue: the degree to which complex sentences pose a challenge for the state-of-the-art ORE systems at the time of writing.

Figure 1 shows a complex sentence annotated with three entities. There are two relations among these entities, namely: ([Kris Knapton]#PER, *spokesperson*, [Metra]#ORG) and ([Metra]#ORG, *located*, [Illinois]#LOC). However, most ORE systems would fail to detect such relations, for a variety of reasons. For instance, the copulas are implicit, rendering useless those extraction patterns that depend on their presence. Also, most extraction systems have a limit on the number of tokens that are allowed between the named en-

tities. On the other hand, most systems would succeed if provided with the following three, much simpler, sentences instead:

- "The pattern is pretty much the same across the nation, said [Kris Knapton]#PER."
- "[Kris Knapton]#PER is a spokesman for [Metra]#ORG."
- "[Metra]#ORG is a commuter rail service in northeastern [Illinois]#LOC."

Throughout the paper, simplified sentences such as the ones above will be referred to as *partial sentences*, as each of them contains only part of the information in the original sentence. The remainder of the paper describes and evaluates effective strategies to re-arrange arbitrary sentences in a way that state-of-the-art ORE systems achieve higher accuracy without introducing a substantial computational penalty.

2. Background and Related Work

NER systems such as those described by Ratinov and Roth (2009) and Finkel et al. (2005) rely on multiple features extracted from the text as well as external knowledge, often called *gazetteers*, to label (primarily) noun phrases as entities in the text. Other techniques are often used to improve NER systems, such as resolving pronouns and abbreviations. Also, such systems often pay special attention to salutations (e.g., "Mr.") and to the capitalization of words. State-of-the-art systems reach very high accuracy on well-formed text such as news articles.

Relation extraction systems build on the output of NER systems. They focus on those sentences containing two or more entities and aim at determining a relation between such entities (if one exists). The first relation extraction systems were specific to a single predefined and domain-specific relation. Systems such as SnowBall (Agichtein and Gravano, 2000) started from known relation instances and learned text patterns to extract previously unknown instances. Several authors employed machine learning for relation extraction; to name just a few, Craven et al. (2000) used many linguistic and statistical features while Bunescu and Mooney (2005) pioneered the use of kernel method. Supervised relation extraction approaches do not scale because the cost in providing training data is linear in the number of relations. Open Relation extraction methods fare better on the Web scenario. Mesquita et al. (2013) study the state-of-the-art in ORE at the time of writing.

To some extent, the method described in this paper bears some resemblance with paraphrasing-based relation extraction methods: they aim at improving relation extraction accuracy by rewriting the sentences, in hopes of arriving at forms that are more amenable to extraction. Romano et al. (2006) explored this idea and proposed a system that assumes the existence of a set of templates that can be used to rewrite the text to arrive at the paraphrases. The authors evaluated the approach on a dataset concerning text about protein interactions, reporting satisfactory results. As for the number of patterns, the authors report that 50% recall can be achieved with 20 patterns or so, while to reach 80% or more recall, hundreds of patters are needed. Unlike this work, the paraphrases generated by their system are not intended to simplify the text.

3. Sentence Re-Structuring

As explained above, ORE systems build on the output of NLP tools that, for instance, annotate the input text with POS tags or word dependencies. Therefore, the effectiveness of an ORE system is limited by the accuracy of the NLP tools themselves. Given the complexity of natural language, such tools are bound to make mistakes, especially when applied to long and complex sentences.

The goal of the method described here is to prevent some of the failures in the NLP tools, by breaking down complex sentences into simpler ones that are easier to process. In turn, these sentences would be easier for existing ORE systems to handle, yielding higher accuracy overall for the relations extracted.

Of course, such decomposition must be done with care, so as to preserve every relation expressed in the original text.

Method Overview. The method proposed in this paper focuses particularly on relative clauses and participle phrases in the original sentences. At a high level, the method works as follows. First, chunking (Jurafsky and Martin, 2008) is applied to break the original sentence into its basic building blocks. The method then determines the relationships among all chunks in the sentence; depending on these relationships, several chunks may be combined together into a partial sentence. However, the chunks are never broken down into separate partial sentences.

More precisely, the method considers chunk C_i and determines if it is *connected*, *disconnected*, or *dependent* on the previous one C_{i-1}. If they are connected, the method joins them. If C_i depends on the previous one, the method creates a new sentence by combing them. If C_i and C_{i-1} are disconnected, the method checks C_i against the last chunk of the previous *partial sentence*. The process is repeated for all chunks.

The next two sections discuss different ways of determining the dependencies of two chunks.

3.1. Sentence Re-Structuring With Parsing

As explained, a crucial task in sentence re-structuring is to determine if two chunks of the original sentence are *connected*, *disconnected*, or *dependent*. One way to if two chunks are related is to use dependency parsing (Jurafsky and Martin, 2008) on the sentence. The dependencies used in this work are the Stanford dependencies for English (Klein and Manning, 2003). Two chunks are said to be *connected* if there exists a word in one chunk with a dependency on a word in the other chunk. Furthermore, if the dependency is rcmod, appos, or partmod, the chunks are said to be *dependent* of each other. If no such dependency exists, the chunks are *disconnected*.

In the example of Figure 1, the chunks "said [Kris Knapton]#PER" and "The pattern is pretty much the same

across the nation" are *connected*; "a spokesman for [Metra]#ORG" and "said [Kris Knapton]#PER" are *dependent*; and "a commuter rail service in northeastern [Illinois]#LOC" and "said [Kris Knapton]#PER" are *disconnected*.

As shown in Section 4., re-structuring via dependency parsing is very effective, in some cases doubling the accuracy of a state-of-the-art ORE system. However, it is also costly, as it requires parsing. Therefore, we also investigated whether we can determine the relationship among chunks using a classifier based on features that do not require parsing.

3.2. Sentence Re-Structuring Without Parsing

This section describes the use of a Naive Bayes classifier, implemented wit the Weka toolkit (Hall et al., 2009), to determine the relationships between chunks at a lower computational cost compared to parsing. The features used by the classifier are: the POS tags of the first 2 and last 2 tokens of each chunk, the chunk tag (NP, VP, etc.), and the distance (in number of tokens) between the chunks.

In order to train the model, distant supervision was used. More precisely, the model was trained with dependencies from 37015 parse trees of The Wall Street Journal section of OntoNotes[1], labeled as *connected*, *disconnected*, or *dependent* according to the criteria above. Note that using parse trees from OntoNotes effectively minimizes potential errors introduced by automatic parsers.

The accuracy of the classifier in a 10-fold cross validation setting is as follows. Overall, 77.7% of the instances are correctly classified. On a per-class basis, the accuracy (f-measure) scores are 0.85 for *disconnected*, 0.75 for *connected* and 0.55 for *dependent*. More importantly, the classifier has much higher precision than recall, leading to a fairly low number of false positives.

4. Experimental Evaluation

This section discusses an experimental evaluation of the sentence re-structuring methods described above with two state-of-the-art ORE systems: ReVerb (Fader et al., 2011) and EXEMPLAR (Mesquita et al., 2013), and three test corpora. These methods were chosen as they constitute the state-of-the-art in open relation extraction, as argued by Mesquita et al. (2013): no other system outperforms these two in terms of both efficiency and accuracy.

ReVerb builds on the premise that most relations are expressed using a few patterns. More precisely, it handles only three types of relations ("verb", "verb+preposition" and "verb+noun+preposition"). Limiting itself to such a small number of patterns, ReVerb requires very little NLP machinery (i.e., it is a "shallow" method), and thus has a very low processing cost per sentence.

EXEMPLAR is a rule-based system that builds on dependencies among terms in the sentence, thus requiring parsing (i.e., it is a "deeper" method). EXEMPLAR detects *triggers* that may indicate relations, and then verifies if there are dependencies in the sentence involving the trigger and two or more named entities. The different rules in the system determine whether or not these dependencies form a relation. It is also worth mentioning that ReVerb is a supervised method while all rules in EXEMPLAR were crafted by hand.

[1]LDC Catalog No LDC2011T03–http://catalog.ldc.upenn.edu/LDC2011T03.

NYT-500 (ground truth: 150 relations)

	P	R	F-1	sec/sent.
ReVerb	0.70	0.11	0.18	0.0146
ReVerb + DEP-SR	0.81	**0.23**	**0.35**	1.1088
ReVerb + NB-SR	**0.82**	0.21	0.33	0.0612
EXEMPLAR	0.72	0.39	0.51	1.0918
EXEMPLAR + DEP-SR	0.74	**0.44**	**0.55**	1.7236
EXEMPLAR + NB-SR	**0.79**	0.41	0.54	0.8954

PENN-100 (ground truth: 51 relations)

	P	R	F-1	sec/sent.
ReVerb	0.78	0.14	0.23	0.0180
ReVerb + DEP-SR	**0.89**	**0.33**	**0.49**	0.6190
ReVerb + NB-SR	0.88	0.29	0.44	0.0670
EXEMPLAR	0.79	0.51	0.62	0.6010
EXEMPLAR + DEP-SR	**0.80**	**0.55**	**0.65**	1.0300
EXEMPLAR + NB-SR	0.76	0.51	0.61	0.5780

WEB-500 (ground truth: 461 relations)

	P	R	F-1	sec/sent.
ReVerb	0.92	0.29	0.44	0.0104
ReVerb + DEP-SR	0.91	0.29	0.44	0.4752
ReVerb + NB-SR	0.91	0.30	0.45	0.0394
EXEMPLAR	0.96	0.46	0.62	0.4862
EXEMPLAR + DEP-SR	0.96	0.46	0.63	0.8940
EXEMPLAR + NB-SR	0.96	0.46	0.63	0.4940

Table 1: Accuracy and performance results on the three test datasets. The columns show the average precision (P), recall (R), f-1 measure (F-1), and time (in seconds) per sentence, for each method.

The methods are compared in terms of precision (P), recall (R), f-measure (F-1) and time (in seconds) per sentence on the test corpora before and after applying the sentence re-structuring method described in this paper. The results of the dependency-based (DEP-SR) and the classifier-based (NB-SR) sentence re-structuring methods are reported separately, for comparison. Table 1 shows all results.

The three corpora, also used in the benchmark of Mesquita et al. (2013), are as follows:

- NYT-500 consists of 500 sentences from the New York Times corpus, manually annotated with binary relations. As shown in Table 1, a total of 150 sentences have relations.
- PENN-100 contains sentences from the Penn Treebank recently used in a tree-kernel ORE method (Xu et al., 2013), where 51 relations are annotated.
- WEB-500 is a commonly used dataset, developed for the TextRunner experiments (Banko and Etzioni, 2008).

Of the three corpora, WEB-500 has the least sophisticated sentences, making it the easier benchmark (as evidenced by the high precision of both ORE systems).

4.1. Discussion

As indicated in Table 1, the dependency-based sentence restructuring method (DEP-SR) increases the effectiveness of both ORE systems in all datasets. The improvements in terms of accuracy are dramatic for ReVerb on the NYT-500 and PENN-100 corpora: 95% increase for NYT-500 and 113% for PENN-100. As for EXEMPLAR, we also observe improvements using DEP-SR, although not nearly as substantial (between 5% and 8%, respectively).

Furthermore, the results in Table 1 provide further evidence that the WEB-500 benchmark is rather simple from an NLP point of view, in the sense that restructuring the sentences has little effect in the accuracy. As expected, the superior accuracy of DEP-SR comes with the added computational cost incurred by parsing the sentences, which is especially noticeable for the case of ReVerb.

Table 1 also shows that NB-SR offers a very attractive compromise, leading to significant accuracy improvements, especially for ReVerb, without increasing the computational cost as much as DEP-SR. Considering the results for EXEMPLAR with our NB-SR method, two observations can be made. First, NB-SR seems to have a small positive effect in increasing the accuracy across all corpora (although a negligible drop in accuracy was observed in the PENN-100 benchmark). Second, NB-SR leads to a *reduction* in processing time, which is due to the fact that it takes EXEMPLAR less time to parse *all* partial sentences produced by the method rather than the original, more complex, one.

5. Conclusion

To the best of the authors knowledge, this work starts the investigation of re-structuring complex sentences to improve relation extraction for arbitrary text. The paper describes a method that breaks the sentences via chunking, and analyzes those chunks to determine which ones should be regrouped together into the same partial sentence. Two strategies for such analysis are presented. One uses dependency parsing and leads to substantial accuracy gains, while the other is based on a classifier that exploits features readily available from the chunks. An experimental evaluation with three sentence corpora, with varying degrees of difficulty, reveals that the method is capable of drastically increasing the accuracy of "shallow" relation extraction systems such as ReVerb or significantly reduce the cost of "deeper" relation extraction systems such as EXEMPLAR.

There are several directions for future work. One would be to further process the chunks and eliminate those that are unlikely to be part of any relation. An immediate approach would be to ignore those partial sentences that do not mention any entities, for example. More generally, the method described here applies only to participal and dependent clauses, and exploiting other sources of complexity may also lead to similar performance gains.

6. Acknowledgements

This work was supported by grants from the Natural Sciences and Engineering Research council of Canada, through its Business Intelligence Network.

7. References

Agichtein, E. and Gravano, L. (2000). Snowball: extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital libraries*, pages 85–94. ACM.

Banko, M. and Etzioni, O. (2008). The tradeoffs between open and traditional relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 28–36. ACL.

Bunescu, R. C. and Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In Mooney, R. J., editor, *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. ACL.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T. M., Nigam, K., and Slattery, S. (2000). Learning to construct knowledge bases from the world wide web. *Artif. Intell.*, 118(1-2):69–113.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying Relations for Open Information Extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. ACL.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. ACL.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing*. Prentice Hall, 2 edition, May.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 423–430. ACL.

Mesquita, F., Schmidek, J., and Barbosa, D. (2013). Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457. ACL.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. ACL.

Romano, L., Kouylekov, M., Szpektor, I., Dagan, I., and Lavelli, A. (2006). Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 409–416. ACL.

Xu, Y., Kim, M.-Y., Quinn, K., Goebel, R., and Barbosa, D. (2013). Open information extraction with tree kernels. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 868–877. ACL.