DOI: http://dx.doi.org/10.14236/ewic/FDIA2015.7

Syntactic and Semantic Structures for Relation Extraction

Duc-Thuan Vo
Laboratory for Systems, Software and Semantics (LS^3)
Ryerson University, Toronto, ON, Canada
ducthuan.vo@ryerson.ca

Ebrahim Bagheri
Laboratory for Systems, Software and Semantics (LS^3)
Ryerson University, Toronto, ON, Canada
bagheri@ryerson.ca

This study proposes to employ syntactic and semantic knowledge from the rich relations within a tree kernel structure for relation extraction. The underlying idea is that different tree kernels with a variety of representations of the available linguistic information will improve the performance of detecting useful pieces of information expressed in a sentence. Applying clause-based rules, clustering algorithms, and bootstrapping on them will help increase the performance of relation extraction. As outlined in this paper, we plan to conduct experiments on recent Information Extraction corpuses and compare the results with the state of the art.

Keywords: Syntactic, Semantic, Tree kernel, Clause-based relation, Clustering algorithms, Bootstrapping

1. INTRODUCTION

Relation extraction (RE) is one of the challenging tasks in information retrieval. The goal of relation extraction is to discover the relevant segments of information in large numbers of textual documents such that they can be used for structuring data. RE aims at discovering various semantic relations in natural language text. It has been applied in many information retrieval tasks such as question answering. For instance answering the question “Who is the President of the United States?” would require a structure where the entity “Barrack Obama” would have the relation “the President of” with another entity “United States”.

Some of the existing research in RE obtains a shallow semantic representation of natural language text in the form of verbs or verbal phrases and their arguments (Bankko et al., 2008; Fader et al., 2011; Wu et al., 2010). Other approaches such as WOEparse (Wu et al., 2010), OLLIE (Mausam et al., 2012), and ClausIE (Corro et al., 2013) use dependency parsing for relation extraction. Each of these approaches makes use of various heuristics to obtain propositions from dependency parsers. Furthermore, bootstrapping (Xu et al., 2007; Xu et al., 2010; Etzioni et al., 2005; Bunescu et al., 2007) has been applied in relation extraction, which does not need a large amount of predefined labels on the training data. It starts from a small set of n-ary relation instances as “seeds”, in order to automatically learn pattern rules from parsed data, which would then be used to extract new instances of relations. Such ER systems learn extraction patterns from dependency trees automatically and systematically induce rules with different complexities. Moreover, several research works have exploited unsupervised methods for relation extraction. They have tried to address this challenge by building on the latent relation hypothesis which states that pairs of words that co-occur in similar contexts tend to have similar relations (Turney, 2008; Rosenfeld et al., 2007; Akbik et al, 2012; Akbik et al., 2014). The authors exploited features using dependency tree to discover relations by clustering entity pairs. Cluster vector space model (pattern) is applied by using the k-mean algorithm and cosine similarity is used to measure distances.

However, existing research face some limitations such as:

1. (P1) Using dependency trees may result in incoherent and uninformative extractions in cases where the extracted relation phrase has no meaningful interpretation. For example, given a sentence “*They recalled that Nungesser began his career as a precinct leader.*", the words *recalled* and *began* are linked together that will create an incoherent relation based on dependency tree-based methods. This will limit maximum recall or may lead to a significant drop of precision at higher points of recall as reported in (Mausam et al., 2012; Wu et al., 2010; Felder et al., 2012; Corro et al., 2013).
2. (P2) Several earlier works such as (Mausam et al., 2012; Wu et al., 2010, Felder et al., 2011) try to apply heuristic rules with Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs)-based sequence labeling for RE. CRF-based approaches are state of the art and they

have yielded high performance in sequence learning tasks. However, the supervised nature of CRF relies on a fairly large amount of training data which must be annotated by humans (Mausam et al., 2012; Corro et al, 2013).

3. (P3) In the work by Xu et al. (Xu et al., 2007; Xu et al., 2010), bootstrapping is applied with predefined rules to train relations based on a dependency tree. However, this approach results in low performance when used on unobserved new domains due to the high likelihood of extracting incorrect rules from the dependency tree during the bootstrapping process.

In order to propose a framework that can address the above three challenges, we have identified the Tree kernel representation to be a solid foundation for our work as it allows us to capture a variety of information including semantic concepts, words, POS tags, shallow and full syntax, dependency parsing, and discourse trees (Xu et al., 2013; Saleh et al., 2014; Zhou et al., 2010; Nguyen et al., 2009; Bunescu et al., 2005). In this study, we will deal with the above three challenges by exploiting the tree kernel structure as follows:

- We will use linguistic knowledge from grammar clauses of the English language to detect relations in rich syntactic and semantic structures for addressing P1. Heuristic rules are applied to obtain proposition relations from the rich tree structure. The rich tree structure includes POS tags, shallow and full syntax, dependency parsing, and discourse trees from the tree kernel that can automatically determine the relations in a sentence. We will use heuristic rules to obtain proposition relations from the rich tree structure.

- In order to address P2, we will model a rich semantic relation tree structure as a vector space model from different tree kernels based on the latent relation hypothesis (Turney, 2008). This representation can compute the similarity of arguments (entity pairs) of relation by comparing the distribution over observed patterns. We then apply clustering methods to find clusters of entity pairs that share similar patterns that can be assumed to represent a relation.

- Finally, we will extend bootstrapping methods by analyzing features from rich syntactic and semantic structures from discourse trees in order to address P3.

2. RELATED WORK

The task of relation extraction was first introduced in the Message Understanding Conference (MUC-6). Since then, a number of techniques have been proposed for this task such as feature vector-based methods and tree kernel-based methods (Xu et al., 2013; Zhou et al., 2010; Nguyen et al., 2009; Bunescu et al., 2005; Vo et al., 2012). Open Information Extraction (OIE) was first presented by Banko et al. (2007) by not being restricted to a pre-specified list of relations in RE. More recent work in Open IE (Akbik, 2009; Wu et al., 2010; Fader et al., 2011; Mausam et al., 2012) have received significant attention. Most of these research work use a shallow semantic representation or dependency parsing in the form of verbs or verbal phrases and their arguments (Banko et al., 2007; Wu et al., 2010; Fader et al., 2011). Mausam et al. (2012) present an improved system called OLLIE, which relaxes the previous systems' constraints that relation words are mediated by verbs, or relation words that appear between two entities. OLLIE creates a training set which includes millions of relations extracted by REVERB (Fader et al., 2011) with high confidence. OLLIE learns relation patterns from the dependency path and lexicon information. Relations that matched the extracted patterns are extracted.

In unsupervised and weakly supervised learning, several authors have built on the latent relation hypothesis which states that pairs of words that co-occur in similar patterns tend to have similar relations (Turney, 2008; Rosenfeld et al., 2007; Akbik et al, 2012; Akbik et al., 2014). These authors exploited features from the dependency tree for discovering relations by clustering entity pairs. Cluster vector space model (pattern) is often applied by using the k-mean algorithm and cosine similarity is used to measure distances. By applying bootstrapping (Xu et al., 2007; Xu et al., 2010; Etzioni et al., 2005; Bunescu et al., 2007), Xu et al., (2007) and Xu et al., (2010) have presented a framework for the extraction of relations. They do not need a large number of predefined labels on the training data. The bootstrapping-based model starts from a small set of n-ary relation instances as "seeds", in order to automatically learn pattern rules from the seed data, which can then extract new relation instances.

As mentioned earlier, the use of dependency trees (Mausam et al., 2012; Corro et al., 2013; Xu et al., 2013) might limit maximum recall or may lead to the drop of precision at higher points of recall due to incoherent and uninformative extractions. Also, RE methods that have employed bootstrapping (Xu et al, 2007; Xu et al., 2010) are limited in their application to new domains due to their focus on relations that are domain specific. We believe that the tree kernel can be a rich syntactic and semantic structure that includes semantic concepts, words, POS tags, shallow and full syntax, dependency parsing and discourse tree (Xu et al., 2013; Saleh et al., 2014; Zhou et al., 2010; Nguyen et al., 2009), which can help to improve the performance when identifying pieces of relation information in a sentence. We suggest that the tree kernel has

potential for improving the performance of ER techniques. Our work aims to augment the tree kernel structure with additional semantic, e.g. named entities concepts and syntactic, e.g. explicit relation nodes (Moschitti, 2006; Zhou et al., 2010; Saleh et al., 2014) for relation extraction as outlined in the following section.

3. OVERVIEW OF THE PROPOSAL APPROACH

The common definition of the RE task is a function from a sentence to a set of triples, such as < *E*1*, R, E*2 >, where *E*1 and *E*2 are entities (noun phrases) and *R* is a relation between the two entities. Several RE systems extract specific relations for prespecified named entity types (Zhou et al., 2010; Nguyen et al., 2009; Bunescu et al., 2005). For instance, *R.MarriedTo(E1.Per, E2.Per)* or *R.LocatedAt(E1.Org, E2.Loc)*. Open Information Extraction (Open IE) (Banko et al., 2007; Corro et al., 2013; Wu and Weld, 2010; Fader et al., 2011; Mausam et al., 2012), a type of RE, aims to extract general relations for two entities. The idea of Open IE is to extract a diverse range of relations and avoid the need for a specific training relation set. For example, *(Tom, married, Marry)* or *(Tom, studies, Computer Science)*. In our work, we propose the following contributions:

3.1. Contribution 1: Tree kernels and clause-based relations

A relation candidate can consist of words before, between, or after the relation pair, or the combination of two consecutive positions. With tree kernel, both learning and classification rely on the inner-product between instances. Tree kernels avoid extracting explicit features from parse trees by calculating the inner product of the two trees, and instead they rely on the common substructure of two trees. We will exploit clauses of the English language to detect relations in rich semantic tree structure. A clause is a part of a sentence that expresses some coherent piece of information; it consists of one subject (S), one verb (V), and optionally an indirect object (O), a direct object (O), a complement (C), and one or more adverbials (A).

Given a sentence "*Obama, the president of the United States, was born in Hawaii on August 4, 1961*", Figure 1 (a) shows the shortest dependency tree path (SDTP) between "*Obama*" and the "*United States*". Additionally, Figure 1 (b) shows a tree kernel with an *R* node added based on the unlexicalized Grammatical Relation Centered Tree (Croce et al. (2011). And, *R* node is as a relation in tree structure. In this example, if a clause structure such as subject-verb-object is considered and *R* is bound to a verb, then relations like *S:Obama; V:the president; O:the United States*, *S:Obama; V:was born; O:Hawaii and S:Obama*; *V:was born; O:on August 4 ,1961* can be extracted.

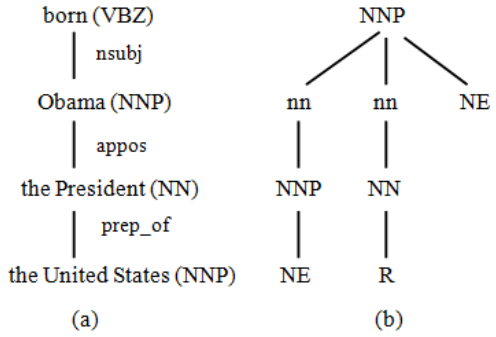

Figure 1: *(a) The shortest dependency tree path (SDTP); (b) Tree structure with "R" added.*

To model the RE problem according to the above example, we will first construct a rich tree structure for a sentence based on the tree kernel. We then gather clauses which exist in the sentence. For each clause, we will determine the set of coherent derived-clauses based on the dependency path, e.g., (*Obama, was born, in Hawaii*) and (*Obama, was born, on August 1961*) from (*Obama, was born*). Finally, we will use heuristics rules to determine, and supervised learning methods such as SVM to classify the proposition relations.

3.2. Contribution 2: Tree kernels and clustering algorithms

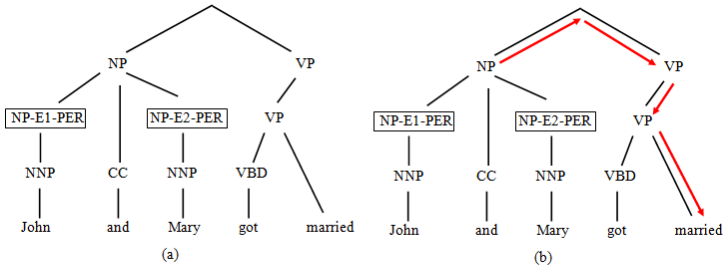

Figure 2: *(a) The shortest dependency tree path (SPT); (b) Predicate-linked: SPT and the rich parse tree structure.*

Current techniques (Turney, 2008; Akbik et al, 2012; Akbik et al., 2014) exploit features from the dependency tree for discovering relations by clustering entity pairs. We choose not to use the dependency path for word extraction due to challenges mentioned above. We will construct a rich semantic-relation tree structure as a vector space model based on different tree-based kernels. We will also discover relations in each sentence by clustering entity pairs. For example, both sentences ''*John and Mary got married.*" and "*John and his wife Mary joined Microsoft*." show the relation *MarriedTo* between entity pairs "*John*" and "*Mary*". We characterize each relation based on a set of common patterns. As an example, Predicate-linked (Figure 2.b) of the sentence "*John and Mary got married*" and tree structure with "R" added (Figure

3.b) of the sentence "*John and his wife Mary joined Microsoft.*" have the similar patterns. The vector space model based on the extracted patterns will follow the latent relation hypothesis (Turney, 2008). We then apply clustering methods to find clusters of entity pairs that share similar patterns representing a specific relation.

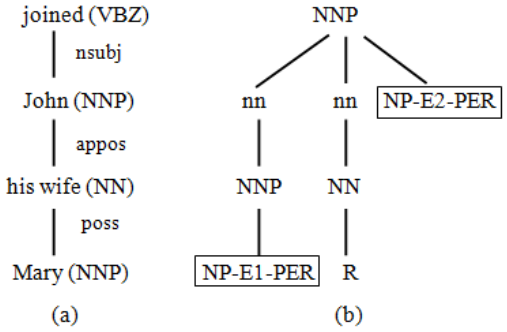

Figure 3: *(a) The shortest dependency tree path (SDTP); (b) Tree structure with "R" added.*

3.3. Contribution 3: Tree kernels and bootstrapping

Xu et al. presented bootstrapping (Xu et al., 2007; Xu et al., 2010) with pre-defined rules to train relations from a dependency tree. Their approach shows low performance in new unobserved domains due to its reliance on a specific corpus. Therefore, in our third contribution, we will use a tree kernel to address the limitations in the dependency tree method through combining it with self-training methods. Our approach will start with some extracted patterns containing potential relations and a small set of relation instances as "seed" in order to train new patterns. The extracted patterns will be based on existing clauses (clauses mentioned in Section 3.1.) in the sentence that will not be limited to a small set of relation types.

For instance, let us consider two relation types *MarriedTo(E1.Per, E2.Per)* and *LocatedAt(E1.Org, E2.Loc)*. The relation *MarriedTo* would need to be associated with two entity pairs *(E1.Person, E2.Person)* and a set of common relation words such as *<"married", "lover", "...">*. Furthermore, the relation *LocatedAt* is associated with entity pairs *(E1.Org, E2.Loc)* and a set of common relation words like *<"located", "is at", "...">*. The self-training methods rely on the RlogF metric whereby those patterns that have more words related with relation instance seeds will receive a higher score (Thelen et al., 2002; Patwardhand et al., 2007). In Figure 4, the extracted patterns P1 and P3 receive high scores for the *MarriedTo* relation and will hence be added to seed of relation *MarriedTo*. The seed of this relation type will be updated with new common words such as "*married*" and "*wife*". Also, the extracted patterns P2 and P6 are added in seed of relation *LocatedAt* with new common word such as "*located at*". Therefore, the system will self improve in the next iteration. By using the self-training method, we will build a new relation list by continuously adding new information to the seeds of the relations.

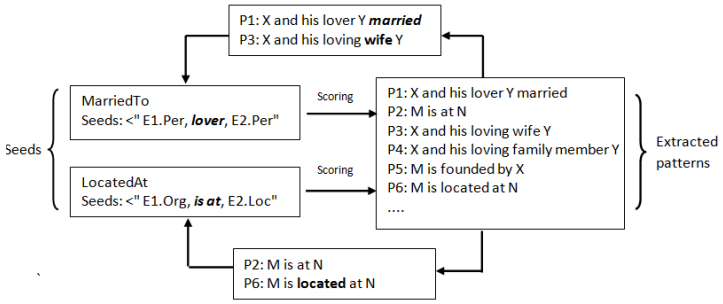

Figure 4: *Self-training process in two relation types.*

4. EVALUATION PLAN

There are three main datasets that are widely used for the evaluation of RE techniques, namely REVERB[1], OLLILE[2] , and ACE[3]. REVERB provides 1,000 tagged training sentences and 500 test sentences. REVERB also provides extracted relations and instance confidence values for the 500 test sentences. OLLIE has a test set which has 300 sentences as well as 900 extracted triples. Finally, ACE RDC 2004 corpus contains 451 documents and 5,702 positive relation instances. It redefines seven entity types, seven major relation types and 23 relation subtypes. Most of the state of the art RE systems perform experiments on these corpuses.

In terms of the state of the art performance, Fader et al., (2011) focus on efficiency by restricting syntactic analysis to part-of-speech tagging and chunking and obtained about precision of 57% and recall of 64% on the REVERB dataset. Mausam et al., (2012) use dependency parsing and various heuristics to obtain propositions relation. They archived around precision of 63% with 600 extracted relations from OLLIE dataset. Corro et al., (2013) also made use of dependency parsing combined with a set of sentence clauses the use various heuristics to obtain propositions from the dependency parses. They archived a precision of 59% with 3,000 extracted relations in REVERB. Xu et al., (2013) proposed multiple SVM models with dependency tree kernels for relation extraction on REVERB and OLLIE datasets, and achieved F-measures of 78.1% in REVERB and 79.3% on OLLIE. Zhou et al., (2010) explored diverse features through a linear kernel and with Support Vector Machines (SVM), and achieved an F-measure of 77.8% in ACE RDC 2004 corpus. Jiang et al., (2007) evaluated the

[1] http://reverb.cs.washington.edu
[2] http://knowitall.github.io/ollie
[3] http://www.ldc.upenn.edu

effectiveness of different feature subspaces with different complexities and obtained the best F-measure of 71.5% on the seven relation types of the ACE RDC 2004 corpus.

We will use these RE corpuses for experiments in our three contributions and compare with state of the art approaches. REVERB and OLLIE will be employed in our first contribution due to not being restricted to a prespecified list of relations. In order to compare with the state of the art such as Xu et al. (2010) and Akbik et al. (2014) in contributions 2 and 3, the ACE RDC 2004 will be used for experiments. We will also use the Stanford parser for analyzing syntactic and semantic structures to be combined with tree kernel (Moschitti et al., 2006).

5. CONCLUDING REMARKS

In this paper, we introduce our proposal for addressing three challenges in RE. We believe that by adding rich syntactic and semantic relation structures to tree kernels, we will be able to improve the state of the art in relation extraction. Our core contribution is to enrich kernel trees with crucial syntactic and semantic information combined with techniques such as clause-based rules, clustering algorithms, and bootstrapping for relation extraction.

ACKNOWLEDGEMENT

We thank ELIAS (Evaluating Information Access System) ESF Research Networking Programme for the awarded scholarship for participating in ESSIR 2015 and FDIA Symposium.

REFERENCES

Akbik, A., Michael, T., Boden, C. (2014) Exploratory Relation Extraction in Large Text Corpora. In Proceedings of *COLING 2014*.

Akbik, A., Visengeriyeva, L.(2012). Unsupervised Discovery of Relation and Discriminative Extraction Patters. In Proceedings of *COLING 2012*.

Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O. (2007). Open Information Extraction from the Web. In Proceedings of *IJCAI 2008.*

Bunescu, R., Mooney, R.J. (2005). Subsequence kernels for relation extraction. In Proceedings of *NIPS 2005*.

Bunescu, R., Mooney, R. (2007). Learning to extract relations from the web using minimal supervision. In Proceedings of *ACL 2007*.

Corro, L.D., Gemulla, R. (2013). ClausIE: Clause-Based Open Information Extraction. In Proceedings of *WWW 2013*.

Croce, D., Moschitti, A., and Basili, R. 2011. Structured lexical similarity via convolution kernels on dependency trees. In Proceedings of *EMNLP 2011*.

Etzioni, O., Cafarella, M., Downey, Doug., Popescu, A.M., Shaked, T., Soderland, S., Weld, D., Yates, Alexander. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, vol. 165, pp. 91 – 134.

Fader, A., Soderland, S., Etzioni, O. (2011) Identifying Relations for Open Information Extraction. In Proceedings of *EMNLP 2011.*

Jiang, J., Zhai, CX. (2007). A systematic exploration of the feature space for relation extraction. In Proceedings of *NAACL-HLT 2007*.

Mausam, Schmitz, M., Bart, R., Soderland, S. (2012). Open Language Learning for Information Extraction. In Proceedings of *EMNLP 2012.*

Moschitti A. (2006) Making tree kernels practical for natural language learning. In Proceedings of *EACL 2006*.

Nguyen, T.V.T, Moschitti, A. (2009). Convolution kernels on constituent, dependency and sequential structures for relation extraction. In Proceedings of the *EMNLP 2009*.

Patwardhan S., Riloff E. (2007). Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In Proceedings of *EMNLP-CoNLL 2007*.

Rosenfeld, B., Feldman, R. (2007). Clustering for unsupervised relation identification. In Proceedings of the *CIKM 2007*.

Saleh, I., Moschitti, A., Nakov, P., Marquez, L., Joty, S. (2014). Semantic Kernels for Semantic Parsing. In Proceedings of *EMNLP 2014*.

Thelen M., Riloff E. (2002). A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. In Proceedings *EMNLP 2002*.

Turney, P. (2008). The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33(1):615–655.

Vo, D. T., Ock, C. Y. (2012). Extraction of Semantic Relation Based on Feature Vector from Wikipedia. In *PRICAI 2012: Trends in Artificial Intelligence*, Springer Berlin Heidelberg.

Wu, F., Weld, D.S. (2010). Open Information Extraction using Wikipedia. In Proceedings of *ACL 2010*.

Xu, F., Uszkoreit, H., Li, H. (2007). A Seed driven Bottom up Machine Learning Framework for

Extracting Relations of Various Complexity. In Proceedings of *ACL 2007*.

Xu, F., Uszkoreit, H., Krause, S., Li, Hong. (2010). Boosting Relation Extraction with Limited Closed-World Knowledge. In Proceedings of *COLING 2010*.

Xu, Y., Kim, M.Y., Quinn, K., Goebel, R., Barbosa, D. (2013). Open Information Extraction with Tree Kernels. In Proceedings of *NAACL-HLT 2013*.

Zhou, G., Qian, L., Fan, J. (2010). Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, vol. 180, 2010.