A Novel Feature-based Approach to Chinese Entity Relation Extraction

Wenjie Li[1], Peng Zhang[1,2], Furu Wei[1], Yuexian Hou[2] and Qin Lu[1]

[1]Department of Computing
The Hong Kong Polytechnic University, Hong Kong
{cswjli,csfwei,csluqin}@comp.polyu.edu.hk

[2]School of Computer Science and Technology
Tianjin University, China
{pzhang,yxhou}@tju.edu.cn

Abstract

Relation extraction is the task of finding semantic relations between two entities from text. In this paper, we propose a novel feature-based Chinese relation extraction approach that explicitly defines and explores nine positional structures between two entities. We also suggest some correction and inference mechanisms based on relation hierarchy and co-reference information etc. The approach is effective when evaluated on the ACE 2005 Chinese data set.

1 Introduction

Relation extraction is promoted by the ACE program. It is the task of finding predefined semantic relations between two entities from text. For example, the sentence "Bill Gates is the chairman and chief software architect of Microsoft Corporation" conveys the ACE-style relation "ORG-AFFILIATION" between the two entities "Bill Gates (PER)" and "Microsoft Corporation (ORG)".

The task of relation extraction has been extensively studied in English over the past years. It is typically cast as a classification problem. Existing approaches include feature-based and kernel-based classification. Feature-based approaches transform the context of two entities into a liner vector of carefully selected linguistic features, varying from entity semantic information to lexical and syntactic features of the context. Kernel-based approaches, on the other hand, explore structured representation such as parse tree and dependency tree and directly compute the similarity between trees. Comparably, feature-based approaches are easier to implement and achieve much success.

In contrast to the significant achievements concerning English and other Western languages, research progress in Chinese relation extraction is quite limited. This may be attributed to the different characteristic of Chinese language, e.g. no word boundaries and lack of morphologic variations, etc. In

this paper, we propose a character-based Chinese entity relation extraction approach that complements entity context (both internal and external) character *N*-grams with four word lists extracted from a published Chinese dictionary. In addition to entity semantic information, we define and examine nine positional structures between two entities. To cope with the data sparseness problem, we also suggest some correction and inference mechanisms according to the given ACE relation hierarchy and co-reference information. Experiments on the ACE 2005 data set show that the positional structure feature can provide stronger support for Chinese relation extraction. Meanwhile, it can be captured with less effort than applying deep natural language processing. But unfortunately, entity co-reference does not help as much as we have expected. The lack of necessary co-referenced mentions might be the main reason.

2 Related Work

Many approaches have been proposed in the literature of relation extraction. Among them, feature-based and kernel-based approaches are most popular.

Kernel-based approaches exploit the structure of the tree that connects two entities. Zelenko et al (2003) proposed a kernel over two parse trees, which recursively matched nodes from roots to leaves in a top-down manner. Culotta and Sorensen (2004) extended this work to estimate similarity between augmented dependency trees. The above two work was further advanced by Bunescu and Mooney (2005) who argued that the information to extract a relation between two entities can be typically captured by the shortest path between them in the dependency graph. Later, Zhang et al (2006) developed a composite kernel that combined parse tree kernel with entity kernel and Zhou et al (2007) experimented with a context-sensitive kernel by automatically determining context-sensitive tree spans.

In the feature-based framework, Kambhatla (2004) employed ME models to combine diverse lexical, syntactic and semantic features derived from word, entity type, mention level, overlap, dependency and parse tree. Based on his work, Zhou et al (2005)

further incorporated the base phrase chunking information and semi-automatically collected country name list and personal relative trigger word list. Jiang and Zhai (2007) then systematically explored a large space of features and evaluated the effectiveness of different feature subspaces corresponding to sequence, syntactic parse tree and dependency parse tree. Their experiments showed that using only the basic unit features within each feature subspace can already achieve state-of-art performance, while over-inclusion of complex features might hurt the performance.

Previous approaches mainly focused on English relations. Most of them were evaluated on the ACE 2004 data set (or a sub set of it) which defined 7 relation types and 23 subtypes. Although Chinese processing is of the same importance as English and other Western language processing, unfortunately few work has been published on Chinese relation extraction. Che et al (2005) defined an improved edit distance kernel over the original Chinese string representation around particular entities. The only relation they studied is PERSON-AFFLIATION. The insufficient study in Chinese relation extraction drives us to investigate how to find an approach that is particularly appropriate for Chinese.

3   A Chinese Relation Extraction Model

Due to the aforementioned reasons, entity relation extraction in Chinese is more challenging than in English. The system segmented words are already not error free, saying nothing of the quality of the generated parse trees. All these errors will undoubtedly propagate to the subsequent processing, such as relation extraction. It is therefore reasonable to conclude that kernel-based especially tree-kernel approaches are not suitable for Chinese, at least at current stage. In this paper, we study a feature-based approach that basically integrates entity related information with context information.

3.1 Classification Features

The classification is based on the following four types of features.
- **Entity Positional Structure Features**
  We define and examine nine finer positional structures between two entities (see Appendix). They can be merged into three coarser structures.
- **Entity Features**
  Entity types and subtypes are concerned.
- **Entity Context Features**
  These are character-based features. We consider both internal and external context. Internal context includes the characters inside two entities and the

characters inside the heads of two entities. External context involves the characters around two entities within a given window size (it is set to 4 in this study). All the internal and external context characters are transformed to Uni-grams and Bi-grams.
- **Word List Features**
  Although Uni-grams and Bi-grams should be able to cover most of Chinese words given sufficient training data, many discriminative words might not be discovered by classifiers due to the severe sparseness problem of Bi-grams. We complement character-based context features with four word lists which are extracted from a published Chinese dictionary. The word lists include 165 prepositions, 105 orientations, 20 auxiliaries and 25 conjunctions.

3.2 Correction with Relation/Argument Constraints and Type/Subtype Consistency Check

An identified relation is said to be correct only when its type/subtype (R) is correct and at the same time its two arguments (ARG-1 and ARG-2) must be of the correct entity types/subtypes and of the correct order. One way to improve the previous feature-based classification approach is to make use of the prior knowledge of the task to find and rectify the incorrect results. Table 1 illustrates the examples of possible relations between PER and ORG. We regard possible relations between two particular types of entity arguments as constraints. Some relations are symmetrical for two arguments, such as PER_SOCIAL.FAMILY, but others not, such as ORG_AFF.EMPLOYMENT. Argument orders are important for asymmetrical relations.

	PER	ORG
PER	PER_SOCIAL.BUS, PER_SOCIAL.FAMILY, …	ORG_AFF.EMPLOYMENT, ORG_AFF.OWNERSHIP, …
ORG		PART_WHOLE.SUBSIDIARY, ORG_AFF.INVESTOR/SHARE, …

Table 1 Possible Relations between ARG-1 and ARG-2

Since our classifiers are trained on relations instead of arguments, we simply select the first (as in adjacent and separate structures) and outer (as in nested structures) as the first argument. This setting works at most of cases, but still fails sometimes. The correction works in this way. Given two entities, if the identified type/subtype is an impossible one, it is revised to NONE (it means no relation at all). If the identified type/subtype is possible, but the order of arguments does not consist with the given relation definition, the order of arguments is adjusted.

Another source of incorrect results is the inconsistency between the identified types and subtypes, since they are typically classified separately.

This type of errors can be checked against the provided hierarchy of relations, such as the subtypes OWNERSHIP and EMPLOYMENT must belong to the ORG_AFF type. There are existing strategies to deal with this problem, such as strictly bottom-up (i.e. use the identified subtype to choose the type it belongs to), guiding top-down (i.e. to classify types first and then subtypes under a certain type). However, these two strategies lack of interaction between the two classification levels. To insure consistency in an interactive manner, we rank the first n numbers of the most likely classified types and then check them against the classified subtype one by one until the subtype conforms to a type. The matched type is selected as the result. If the last type still fails, both type and subtype are revised to NONE. We call this strategy type selection. Alternatively, we can choose the most likely classified subtypes, and check them with the classified type (i.e. subtype selection strategy). Currently, n is 2.

3.2 Inference with Co-reference Information and Linguistic Patterns

Each entity can be mentioned in different places in text. Two mentions are said to be co-referenced to one entity if they refers to the same entity in the world though they may have different surface expressions. For example, both "he" and "Gates" may refer to "Bill Gates of Microsoft". If a relation "ORG-AFFILIATION" is held between "Bill Gates" and "Microsoft", it must be also held between "he" and "Microsoft". Formally, given two entities $E1=\{EM_{11}, EM_{12}, …, EM_{1n}\}$ and $E2=\{EM_{21}, EM_{22}, …, EM_{2m}\}$ (E_i is an entity, EM_{ij} is a mention of E_i), it is true that $R(EM_{11}, EM_{21}) \Rightarrow R(EM_{1l}, EM_{2k})$. This nature allows us to infer more relations which may not be identified by classifiers.

Our previous experiments show that the performance of the nested and the adjacent relations is much better than the performance of other structured relations which suffer from unbearable low recall due to insufficient training data. Intuitively we can follow the path of "Nested $\Rightarrow$ Adjacent $\Rightarrow$ Separated $\Rightarrow$ Others" (Nested, Adjacent and Separated structures are majority in the corpus) to perform the inference. But soon we have an interesting finding. If two related entities are nested, almost all the mentions of them are nested. So basically inference works on "Adjacent $\Rightarrow$ Separated".

When considering the co-reference information, we may find another type of inconsistency, i.e. the one raised from co-referenced entity mentions. It is possible that $R(EM_{11}, EM_{21}) \neq R(EM_{12}, EM_{22})$ when R

is identified based on the context of EM. Co-reference not only helps for inference but also provides the second chance to check the consistency among entity mention pairs so that we can revise accordingly. As the classification results of SVM can be transformed to probabilities with a sigmoid function, the relations of lower probability mention pairs are revised according to the relation of highest probability mention pairs.

The above inference strategy is called coreference-based inference. Besides, we find that pattern-based inference is also necessary. The relations of adjacent structure can infer the relations of separated structure if there are certain linguistic indicators in the local context. For example, given a local context "EM_1 and EM_2 located EM_3", if the relation of EM_2 and EM_3 has been identified, EM_1 and EM_3 will take the relation type/subtype that EM_2 and EM_3 holds. Currently, the only indicators under consideration are "and" and "or". However, more patterns can be included in the future.

4   Experimental Results

The experiments are conducted on the ACE 2005 Chinese RDC training data (with true entities) where 6 types and 18 subtypes of relations are annotated. We use 75% of it to train SVM classifiers and the remaining to evaluate results.

The aim of the first set of experiments is to examine the role of structure features. In these experiments, a "NONE" class is added to indicate a null type/subtype. With entity features and entity context features and word list features, we consider three different classification contexts: (1), only three coarser structures[1], i.e. nested, adjacent and separated, are used as feature, and a classifier is trained for each relation type and subtype; (2) similar to (1) but all nine structures are concerned; and (3) similar to (2) but the training data is divided into 9 parts according to structure, i.e. type and subtype classifiers are trained on the data with the same structures. The results presented in Table 2 show that 9-structure is much more discriminative than 3-structure. Also, the performance can be improved significantly by dividing training data based on nine structures.

Type / Subtype	Precision	Recall	F-measure
3-Structure	0.7918/0.7356	0.3123/0.2923	0.4479/0.4183
9-Structure	0.7533/0.7502	0.4389/0.3773	0.5546/0.5021
9-Structure_Divide	0.7733/0.7485	0.5506/0.5301	0.6432/0.6209

Table 2 Evaluation on Structure Features

Structure	Positive Class	Negative Class	Ratio
Nested	6332	4612	1 : 0.7283
Adjacent	2028	27100	1 : 13.3629

[1] Nine structures are combined to three by merging (b) and (c) to (a), (e) and (f) to (d), (h) and (i) to (g).

Separated	939	79989	1 : 85.1853
Total	9299	111701	1 : 12.01

Table 3 Imbalance Training Class Problem

In the experiments, we find that the training class imbalance problem is quite serious, especially for the separated structure (see Table 3 above where "Positive" and "Negative" mean there exists a relation between two entities and otherwise). A possible solution to alleviate this problem is to detect whether the given two entities have some relation first and if they do then to classify the relation types and subtypes instead of combining detection and classification in one process. The second set of experiment is to examine the difference between these two implementations. Against our expectation, the sequence implementation does better than the combination implementation, but not significantly, as shown in Table 4 below.

Type / Subtype	Precision	Recall	F-measure
Combination	0.7733/0.7485	0.5506/0.5301	0.6432/0.6206
Sequence	0.7374/0.7151	0.5860/0.5683	0.6530/0.6333

Table 4 Evaluation of Two Detection and Classification Modes

Based on the sequence implementation, we set up the third set of experiments to examine the correction and inference mechanisms. The results are illustrated in Table 5. The correction with constraints and consistency check is clearly contributing. It improves F-measure 7.40% and 6.47% in type and subtype classification respectively. We further compare four possible consistency check strategies in Table 6 and find that the strategies using subtypes to determine or select types perform better than top down strategies. This can be attributed to the fact that correction with relation/argument constraints in subtype is tighter than the ones in type.

Type / Subtype	Precision	Recall	F-measure
Seq. + Cor.	0.8198/0.7872	0.6127/0.5883	0.7013/0.6734
Seq. + Cor. + Inf.	0.8167/0.7832	0.6170/0.5917	0.7029/0.6741

Table 5 Evaluation of Correction and Inference Mechanisms

Type / Subtype	Precision	Recall	F-measure
Guiding Top-Down	0.7644/0.7853	0.6074/0.5783	0.6770/0.6661
Subtype Selection	0.8069/0.7738	0.6065/0.5817	0.6925/0.6641
Strictly Bottom-Up	0.8120/0.7798	0.6146/0.5903	0.6996/0.6719
Type Selection	0.8198/0.7872	0.6127/0.5883	0.7013/0.6734

Table 6 Comparison of Different Consistency Check Strategies

Finally, we provide our findings from the fourth set of experiments which looks at the detailed contributions from four feature types. Entity type features themselves do not work. We incrementally add the structures, the external contexts and internal contexts, Uni-grams and Bi-grams, and at last the word lists on them. The observations are: Uni-grams provide more discriminative information than Bi-grams; external context seems more useful than internal context; positional structure provides stronger support than other individual recognized features such as entity type and context; but word list feature can not further boost the performance.

Type / Subtype	Precision	Recall	F-measure
Entity Type + Structure	0.7288/0.6902	0.4876/0.4618	0.5843/0.5534
+ External (Uni-)	0.7935/0.7492	0.5817/0.5478	0.6713/0.6321
+ Internal (Uni-)	0.8137/0.7769	0.6113/0.5836	0.6981/0.6665
+ Bi- (Internal & External)	0.8144/0.7828	0.6141/0.5902	0.7002/0.6730
+ Wordlist	0.8167/0.7832	0.6170/0.5917	0.7029/0.6741

Table 6 Evaluation of Feature and Their Combinations

5 Conclusion

In this paper, we study feature-based Chinese relation extraction. The proposed approach is effective on the ACE 2005 data set. Unfortunately, there is no result reported on the same data so that we can compare.

6 Appendix: Nine Positional Structures
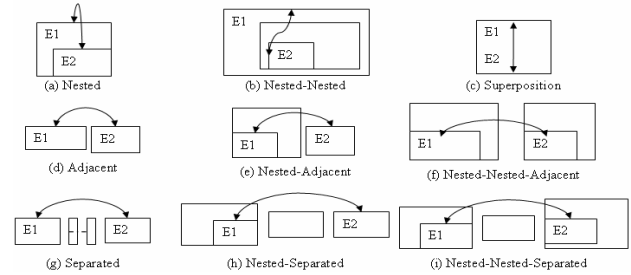
Acknowledgments

References

Razvan Bunescu and Raymond Mooney. 2005. A Shortest Path Dependency Tree Kernel for Relation Extraction, In Proceedings of HLT/EMNLP, pages 724-731.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency Tree Kernels for Relation Extraction, in Proceedings of ACL, pages 423-429.

Jing Jiang, Chengxiang Zhai. 2007. A Systematic Exploration of the Feature Space for Relation Extraction. In proceedings of NAACL/HLT, pages 113-120.

Nanda Kambhatla. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. In Proceedings of ACL, pages 178-181.

Dmitry Zelenko, Chinatsu Aone and Anthony Richardella. 2003. Kernel Methods for Relation Extraction. Journal of Machine Learning Research 3:1083-1106

Min Zhang, Jie Zhang, Jian Su and Guodong Zhou. 2006. A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features, in Proceedings of COLING/ACL, pages 825-832.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring Various Knowledge in Relation Extraction. In Proceedings of ACL, pages 427-434.

GuoDong Zhou, Min Zhang, Donghong Ji and Qiaoming Zhu. 2007. Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. In Proceedings of EMNLP, pages 728-736.

Wanxiang Che et al. 2005. Improved-Edit-Distance Kernel for Chinese Relation Extraction. In Proceedings of IJCNLP, pages 132-137.