Exploiting the Role of Position Feature in Chinese Relation Extraction

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

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]School of Computer Science and Technology, Tianjin University, China

E-mail: {cspzhang, cswjli, csfwei, csluqin}@comp.polyu.edu.hk, yxhou@tju.edu.cn

Abstract

Relation extraction is the task of finding pre-defined semantic relations between two entities or entity mentions from text. Many methods, such as feature-based and kernel-based methods, have been proposed in the literature. Among them, feature-based methods draw much attention from researchers. However, to the best of our knowledge, existing feature-based methods did not explicitly incorporate the position feature and no in-depth analysis was conducted in this regard. In this paper, we define and exploit nine types of position information between two named entity mentions and then use it along with other features in a multi-class classification framework for Chinese relation extraction. Experiments on the ACE 2005 data set show that the position feature is more effective than the other recognized features like entity type/subtype and character-based *N*-gram context. Most important, it can be easily captured and does not require as much effort as applying deep natural language processing.

1. Introduction

The research in relation extraction was promoted by the Message Understanding Conferences (MUCs) (MUC, 1987-1998) and the Automatic Content Extraction (ACE) program (ACE, 2002-present). According to the ACE program, an entity is an object or a set of objects in the world and a relation is an explicitly or implicitly stated relationship among entities. For example, the sentence "George Bush traveled to France on Thursday for a summit" conveys the ACE-style relation "Physical. Located" between the two entities "George Bush (Person)" and "France (Location)", where "Physical" and "Located" are the pre-defined relation type and subtype.

In text, an entity may have more than one entity mention. Mentions are co-referent with each other and inherit the entity type and subtype from the corresponding entity they belong to. For example, "George W. Bush" and "the president of the United States" are two different mentions with different linguistic expressions but they refer to the same person and belong to the same person entity. Extraction of semantic relations between entities or entity mentions can be very useful in many NLP applications, such as information extraction, question answering and ontology construction.

In general, the task of relation extraction is to decide the semantic relations between two entities (or entity mentions[1]) in the context (e.g. in a sentence, or a small piece of text). Since relation types and subtypes are predefined, this task is usually modeled as a classification problem. Many methods, such as feature-based methods (Kambhatla 2004; Zhou et al 2005) and kernel-based methods (Zelenko et al. 2003; Culotta and Sorensen, 2004; Zhang et al 2006; Zhou et al 2007), have been proposed in literature. In this paper, we are particularly interested in feature-based methods.

Kambhatla (2004) employed Maximum Entropy models with features derived from word, entity type, mention level, overlap, dependency tree and parse tree. Zhou et al (2005) further incorporated the base phrase chunking information. The above two works both adopted overlap features, which implicitly reflect the position feature of two entity mentions. Jiang and Zhai (2007) systematically explored a large space of features for relation extraction and evaluated the effectiveness of different feature subspaces. They concluded that using basic unit features was generally sufficient to achieve state-of-art performance, while over-inclusion of complex features might hurt the performance.

The feature-based methods draw much attention from researchers. However, to the best of our knowledge, existing feature-based methods did not explicitly incorporate the position feature, which can be very useful in our observations. This motivates us to further study the position information between two named entity mentions for Chinese relation extraction. Experiments on the ACE 2005 data set show that the position feature can be more effective than the other recognized features like entity type /subtype and character-based *N*-gram context. Meanwhile it can be easily captured with less effort than applying deep natural language processing.

The rest of this paper is organized as follows. Section 2 describes three kinds of classification features (especially the position feature). Experimental studies are then presented in Section 3. This is followed by discussion in Section 4 and conclusion in Section 5.

[1] In this paper, we consider the relations between two entity mentions.

2. Classification-based Chinese Relation Extraction

In this paper, the task of relation extraction is modeled as a classification problem. Section 2 first describes three kinds of features involved, and then presents the vector representations of these features used as the input to the classification tools.

2.1. Features for Classification

2.1.1. Position Feature

We define the position feature (including nine position types) between two named entity mentions as follows:

Given a named entity mention *nem*, let *nem.start* and *nem.end* denote the start and end positions of *nem* in a sentence respectively. Let $nem_i \supset nem_j$ denotes $(nem_i.start,\ nem_i.end) \supset (nem_j.start,\ nem_j.end)$ and $(nem_i.start,\ nem_i.end) \neq (nem_j.start,\ nem_j.end)$, and let $nem_k \perp (nem_1, nem_2)$ denotes $nem_1.end < nem_k.start$ and $nem_k.end < nem_2.start$. For any two entity mentions nem_1 and nem_2, where $nem_1 \supset nem_2$ or nem_1 precedes nem_2, the position of them can be completely grouped into nine types, as illustrated in Table 1 below.

Type	Condition	Label
Nested	$nem_1 \supset nem_2 \wedge \neg\exists (nem_i)(nem_1 \supset nem_i \wedge nem_i \supset nem_2)$	(a)
Nested-Nested	$Nem_1 \supset nem_2 \wedge \exists (nem_i)(nem_1 \supset nem_i \wedge nem_i \supset nem_2)$	(b)
Superposition	$nem_1.start = nem_2.start$ and $nem_1.end = nem_2.end$	(c)
Adjacent	$nem_1.end < nem_2.start \wedge \neg\exists (nem_i)(nem_i \supset nem_1 \vee nem_i \supset nem_2) \wedge \neg\exists (nem_j)(nem_j \perp (nem_1, nem_2))$	(d)
Nested-Adjacent	$nem_1.end < nem_2.start \wedge (\exists (nem_i)(nem_i \supset nem_1 \wedge \neg\exists (nem_j)(nem_j \supset nem_2)) \vee \exists (nem_i)(nem_i \supset nem_2 \wedge \neg\exists (nem_j)(nem_j \supset nem_1))) \wedge \neg\exists (nem_j)(nem_j \perp (nem_1, nem_2))$	(e)
Nested-Nested-Adjacent	$nem_1.end < nem_2.start \wedge \exists (nem_i)(nem_i \supset nem_1) \wedge \exists (nem_j)(nem_j \supset nem_2) \wedge \neg\exists (nem_j)(nem_j \perp (nem_1, nem_2))$	(f)
Separated	$\exists (nem_j)(nem_j \perp (nem_1, nem_2)) \wedge \neg\exists (nem_i)(nem_i \supset nem_1 \vee nem_i \supset nem_2)$	(g)
Nested-Separated	$\exists (nem_j)(nem_j \perp (nem_1, nem_2))(\exists (nem_i)(nem_i \supset nem_1 \wedge \neg\exists (nem_j)(nem_j \supset nem_2)) \vee \exists (nem_i)(nem_i \supset nem_2 \wedge \neg\exists (nem_j)(nem_j \supset nem_1)))$	(h)
Nested-Nested-Separated	$\exists (nem_j)(nem_j \perp (nem_1, nem_2)) \wedge \exists (nem_i)(nem_i \supset nem_1) \wedge \exists (nem_j)(nem_j \supset nem_2)$	(i)

Table 1. Nine positions between two named entity mentions (see Appendix also)

2.1.2. Entity Feature

This feature concerns the entity type and subtype of two named entity mentions.

2.1.3. *N*-gram Context Feature

The context features concern characters around two named entity mentions in a given window size *w_s*. The characters can be classified into the following four types:

- CBM1: at most *w_s* characters before nem_1
- CAM1: at most *w_s* characters after nem_1
- CBM2: at most *w_s* characters before nem_2
- CAM2: at most *w_s* characters after nem_2

The extraction of the above characters must comply with two rules. First, these characters can not cross any adjacent entity mention. Second, if there is another name entity mentions nem_i contains nem_1 (or nem_2), these characters can not cross the borders of nem_i, i.e., characters must be inside nem_i. Notice that we use the characters instead of the words considering Chinese word-based models can be heavily affected by word segmentation errors.

2.2. The Classification Tool

Support Vector Machine (SVM) is selected as the classification tool, considering it represents the state-of-the-art in the machine learning research community, and good implementations of the algorithm are available.

2.3. Vector Representation for SVM

As described in (Manevitz and Yousef 2001), there are four different text representations, i.e., binary, frequency, *tf-idf*, and Hadamard. In this paper, we apply binary vector representation to the features extracted. Since each feature has its own characteristic, we describe the vector representation of each feature as follows.

2.3.1. Representation of Position Feature

For this feature, we choose to use the 9-dimensional binary vector where the i^{th} entry is 1 if the position is the i^{th} type, and the other entries are 0.

2.3.2. Representation of Entity Feature

Supposing the numbers of the entity type and subtype are *n_type* and *m_subtype* respectively, we need to choose two binary vectors (*n_type*-dimensional and *m_subtype*-dimensional) to represent the type and subtype of a given named entity mention. The i^{th} entry of the corresponding vector is 1 if the i^{th} type or subtype is encountered.

2.3.3. Representation of *N*-gram Context Feature

Only Uni-gram is considered as the *N*-gram feature. Suppose the total number of the Uni-grams in the corpus is *n_uni_grams*. For each character sequence, a *n_uni_grams*-dimensional vector is chosen to represent the corresponding uni-gram feature. The i^{th} entry of the corresponding vector is 1 if the i^{th} uni-gram appears in the given character sequence.

3. Experimental Studies

3.1. Experiment Set-up

The experiment is set up on the training data set of the ACE 2005 Chinese Relation Detection and Characterization task provided by the Linguistic Data Consortium[2]. The 633 documents have been manually annotated with 9299 instances of relations. 6 relation types and 18 subtypes are pre-defined. More detail information is shown in Table 2. Because of no test data at hand, we randomly select 474 out of the 633 documents (i.e. 75%) as the training data and the remaining documents are used for evaluation.

Relation Type	Relation Subtype	Frequency
ART (Total No: 630)	User-Owner-Inventor -Manufacturer	630
GEN-AFF (Total No: 1937)	Citizen-Resident-Religion -Ethnicity	746
	Org-Location	1191
ORG-AFF (Total No: 2198)	Employment	1584
	Founder	17
	Ownership	25
	Student-Alum	72
	Sports-Affiliation	69
	Investor-Shareholder	85
	Membership	346
PART-WHOLE (Total No: 2286)	Artifact	14
	Geographical	1289
	Subsidiary	983
PER-SOC (Total No: 660)	Business	188
	Family	384
	Lasting-Personal	88
PHYS(Physical) (Total No: 1588)	Located	1358
	Near	230

Table 2. Relation types and subtypes in the ACE 2005 training corpus.

We first extract the three types of features mentioned in Section 3, and then adopt SVMlight (Joachims 1998) as the multi-class classification tool. In our experiments, the window size for context feature extraction is set to 4 characters around the entity mentions and the contextual characters can not across any named entity mentions. Linear kernel is used and the training parameter C is set to 5000. Two classifiers, a 7-class and a 19-class classifier are trained independently to predict the relation types and subtypes respectively. For both classifiers, we add a "NONE" class when the two relation mentions are not related (i.e. no relation between them).

3.2. Experimental Results

The aim of the first set of experiments is to examine the performance of the position feature. Table 3 and Table 4 below report the precision, recall and F-score results of the three features and their incremental combinations on relation type detection and recognition (RTDR) and relation subtype detection and recognition (RSDR). We come up with the following observations and conclusions. First, the performances are extremely bad when the three features are used individually, although the position feature performs the best. Second, when the two features are combined in use, the performance is already competing as long as the position feature is involved. Finally, the best performance is achieved when all of the three features are integrated.

	Precision	Recall	F-measure
Entity	0	0	0
Context	0.448598	0.0229226	0.0436165
Position	0.207921	0.190544	0.198854
Entity + Context	0	0	0
Entity + Position	0.654581	0.474212	0.549986
Context + Position	0.481356	0.271251	0.346976
Entity + Context + Position	0.70126	0.457767	0.553937

Table 3 Comparison of 7 different feature spaces over relation types in the test data set

	Precision	Recall	F-measure
Entity	0	0	0
Context	0.440367	0.0229226	0.0435769
Position	0.025013	0.0229226	0.0239223
Entity + Context	0	0	0
Entity + Position	0.659142	0.418338	0.511832
Context + Position	0.455696	0.240688	0.315
Entity + Context + Position	0.677718	0.419771	0.518431

Table 4. Comparison of 7 different feature spaces over relation subtypes in the test data set

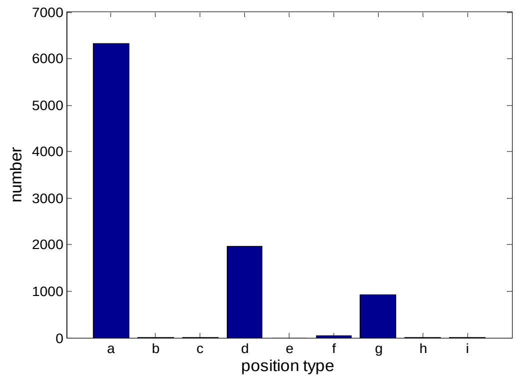


Figure 1. The number of relation instances of every position types

As shown form the distribution of relation instances of 9 position types (see Figure 1 above), the numbers of relation instances of position (a) (d) and (g) are significantly more than the numbers of the other positions. So we map 9-Position to 3-Position to examine the influence of the position feature. The mapping strategy is: (b) and (c) are mapped to (a), (e) and (f) are mapped to (d), (h) and (i) are mapped to (g). Table 5 and Table 6 show the comparison of 3-Position feature and 9-Position feature. It

[2] http://www.ldc.upenn.edu/Projects/ACE

shows that 9-Position feature outperforms 3-Position feature by 10.5% of F-measure for RTDR and 9.2% of F-measure for RSDR when incorporating with the other two features.

	Precision	**Recall**	**F-measure**
Entity + Context + 9-Position	0.70126	0.457767	0.553937
Entity + Context + 3-Position	0.637885	0.34575	0.448436

Table5. Comparison of 9-Position and 3-Position over relation types in the test dataset

	Precision	**Recall**	**F-measure**
Entity + Context + 9-Position	0.677718	0.419771	0.518431
Entity + Context + 3-Position	0.668024	0.313276	0.426528

Table 6. Comparison of 9-Position and 3-Position over relation subtypes in the test dataset

4. Discussion on 9-Position and 3-Position

We discuss the class imbalance problem of 9-position and 3-position in the task of relation extraction.

The class imbalance problem typically occurs when, in a classification problem, there are many more instances of some classes than others. In such cases, standard classifiers tend to overwhelmed by large classes and ignore the small ones, and then cause a significant bottleneck in performance (Japkowicz 2000 and Chawla et al. 2004).

Unfortunately, the task of relation extraction encounter imbalance problems (Culotta et al. 2006 and Kambhatla 2006), i.e., there are many more "NONE" (negative) class relation instances than predefined (positive) classes. As we can be seen from Tables 7 and 8, the ratio of positive to negative class on the whole ACE corpus is 1: 12.01.

Position Type	**Positive Class**	**Negative Class**	**Ratio**
Nested	6325	2347	1 : 0.37
Adjacent	1978	13501	1 : 6.82
Separated	928	39808	1 : 42.87
Nested-Nested	1	1858	1 : 1858
Nested-Adjacent	0	10119	1 : INF
Nested-Separated	1	31039	1 : 31039
Nested-Nested-Adjacent	50	3480	1 : 69.6
Nested-Nested-Separated	10	9142	1 : 914.2
Superposition	6	407	1 : 67.84
Total	9299	111701	1 : 12.01

Table 7. The ratios of positive to negative class on 9-Position types

Position Type	**Positive Class**	**Negative Class**	**Ratio**
Nested	6332	4612	1 : 0.7283
Adjacent	2028	27100	1 : 13.3629
Separated	939	79989	1 : 85.1853
Total	9299	111701	1 : 12.01

Table 8 The ratios of positive to negative class on merged 3-Position types

As shown in Section 3, 9-Position outperforms 3-Position. This can be attributed to the fact that 9-Position is more discriminative than 3-Position, and the imbalance problem of the three main positions (i.e., Nested, Adjacent, Separated) in 3-Position is much worse than the one in 9-Position.

5. Conclusion

In this paper, we study the role of the position feature in Chinese relation extraction. Nine types of position information between two named entity mentions are defined and then used as one of the features in relation classification. Experiments on the ACE 2005 data set show that the position feature is quite effective.

6. Acknowledgements

This work was supported in part by the HK Research Grants Council (Grant No. CERG PolyU 5211/05E), Natural Science Foundation of China (Grant No. 60603027), and Applied Basic Research Project of Tianjin of China (Grant No. 05YFJMJC11700).

7. References

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

Culotta A. and Sorensen J. (2004). Dependency Tree Kernels for Relation Extraction, in Proceedings of ACL, pages 423-429.

Jiang J. and Zhai C. (2007). A Systematic Exploration of the Feature Space for Relation Extraction. In proceedings of NAACL/HLT, pages 113-120.

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

Zelenko D, Aone C. and Richardella A. (2003). Kernel Methods for Relation Extraction. Journal of Machine Learning Research 3:1083-1106

Zhou G., Su J., Zhang J., and Zhang M. (2005). Exploring Various Knowledge in Relation Extraction. In Proceedings of ACL, pages 427-434.

Zhang M., Zhang J., Su J. and Zhou G. (2006). A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features, in Proceedings of COLING/ACL, pages 825-832.

Joachims T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. *In Proceedings of European Conference on Machine Learning*.

Manevitz L. M. and Yousef M. (2001) One-class svms for document classification. Journal of Machine Learning Research, 2(Dec):139-154

Zhou G., Zhang. M, Ji D. and Zhu Q. (2007). Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. In Proceedings of EMNLP, pages 728-736.

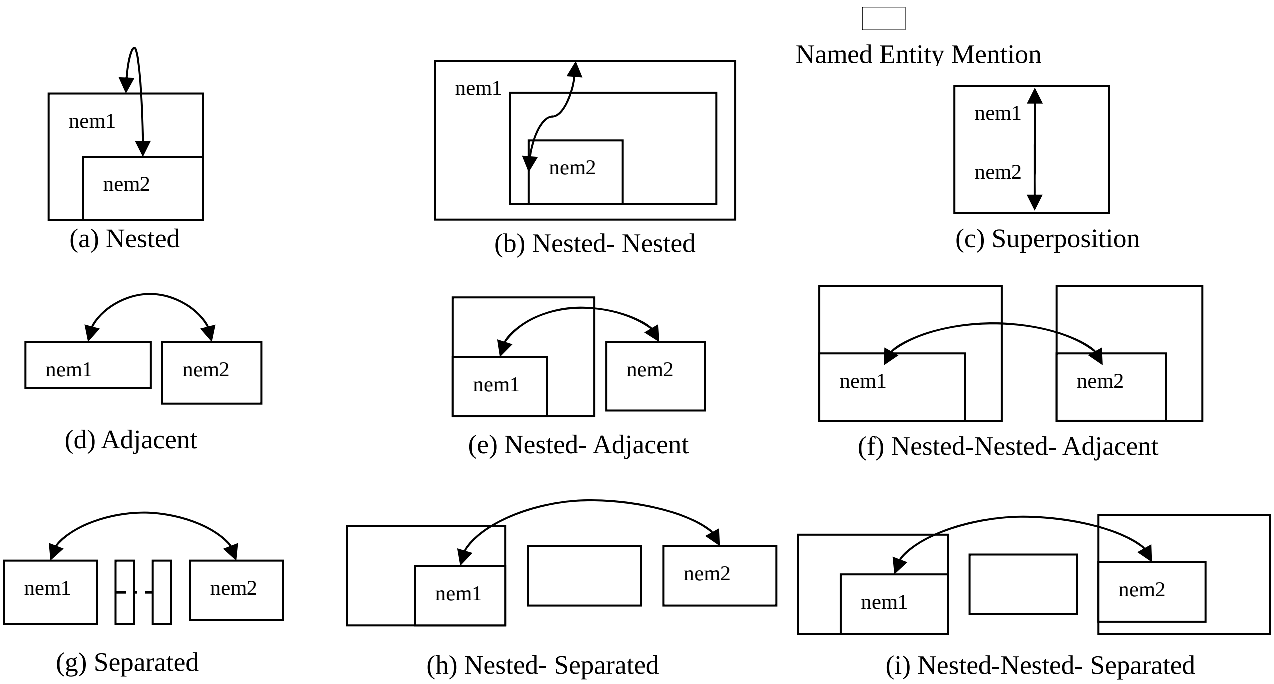
Chawla N., Japkowicz N. and Kolcz A. (2004) "Editorial: Special Issue on Learning from Imbalanced Datasets," *SIGKDD Explorations*, 6(1):1-6.

Japkowicz, N. (2000). Learning from imbalanced data sets: a comparison of various strategies. In Papers from the AAAI Workshop on Learning from Imbalanced Data Sets. Tech. rep. WS-00-05, Menlo Park, CA: AAAI Press.

Culotta A. McCallum A, Betz J.(2006) Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text. HLT-NAACL, pages 296-303

Kambhatla N.(2006) Minority Vote: At-Least-N Voting Improves Recall for Extracting Relations. In Proceedings of ACL, pages 460-466.

Appendix. Nine Position Types



(a) Nested

(b) Nested- Nested

(c) Superposition

(d) Adjacent

(e) Nested- Adjacent

(f) Nested-Nested- Adjacent

(g) Separated

(h) Nested- Separated

(i) Nested-Nested- Separated