Developing Position Structure based Framework for Chinese Entity Relation Extraction

PENG ZHANG
The Robert Gordon University
WENJIE LI
The Hong Kong Polytechnic University
YUEXIAN HOU
Tianjin University
AND
DAWEI SONG
The Robert Gordon University

Relation extraction is the task of finding semantic relations between two entities in text, and is often cast as a classification problem. In contrast to the significant achievements on English language, research progress in Chinese relation extraction is relatively limited. In this paper, we present a novel Chinese relation extraction framework, which is mainly based on a 9-position structure. The design of this proposed structure is motivated by the fact that there are some obvious connections between relation types/subtypes and position structures of two entities. The 9-position structure can be captured with less effort than applying deep natural language processing, and is effective to relieve the class imbalance problem which often hurts the classification performance. In our framework, all involved features do not require Chinese word segmentation, which has long been limiting the performance of Chinese language processing. We also utilize some correction and inference mechanisms to further improve the classified results. Experiments on the ACE 2005 Chinese data set show that the 9-position structure feature can provide strong support for Chinese relation extraction. As well as this, other strategies are also effective to further improve the performance.

Categories and Subject Descriptors: I 2.7 [**Artificial Intelligence**] – Natural language processing – *Text Analysis*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Entity relation extraction, Chinese language, position structure, imbalance class classification

1. INTRODUCTION

Relation extraction is a task to find semantic relations between two entities from the text. This task is recently promoted by the Automatic Content Extraction (ACE) Evaluation program. For instance, the sentence "Bill Gates is the chairman of Microsoft

---

The majority of this work was done when P. Zhang was a research assistant at The Hong Kong Polytechnic University. This paper is an extended version of an ACL 2008 article [Li et al. 2008].

Authors' addresses: P. Zhang and D. Song, School of Computing, The Robert Gordon University. E-mail: {p.zhang1, d.song}@rgu.ac.uk. W. Li, Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: cswjli@comp.polyu.edu.hk; Y. Hou, School of Computer Science and Technology, Tianjin University, China. E-mail: yxhou@tju.edu.cn

Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Permission may be requested from the Publications Dept., ACM, Inc., 2 Penn Plaza, New York, NY 11201-0701, USA, fax: +1 (212) 869-0481, permission@acm.org
© 2001 ACM 1530-0226/07/0900-ART9 $5.00

Corporation” conveys the ACE-style relation “ORG-AFFILIATION” between the two entities “Bill Gates (PER)” and “Microsoft Corporation (ORG)”, where PER and ORG are entity types, and ORG-AFFILIATION is a relation type.

The task of relation extraction has been extensively studied over the past years mainly for English. It is usually cast as a classification problem. Existing approaches include feature-based and kernel-based methods. Feature-based approaches [Kambhatla 2004; Zhou et al. 2005; Jiang and Zhai 2007; Zhou et al. 2009] transform the context of two entities into a linear vector of carefully selected linguistic features, varying from entity semantic information to lexical and syntactic features of the context. Kernel-based approaches [Zhang et al. 2006; Zhou et al. 2007a, 2010], on the other hand, design kernel functions on the relation context’s structured representation such as parse tree or dependency tree, and then compute the similarity between two relation instances.

In contrast to the significant achievement concerning English and other Western languages, research progress in Chinese relation extraction is relatively limited. This might be due to the nature of Chinese language, e.g. no word boundaries and lack of morphological variations, etc. The system-segmented words are already not error free, thus also affecting the quality of the generated parse trees. All these errors will undoubtedly propagate to a subsequent processing, such as the relation extraction. It is therefore reasonable to conclude that word-based features and kernel-based (especially tree-kernel-based) approaches are not suitable for Chinese, at least at the current stage. Huang et al. [2008] provided empirical evidence showing that in ACE 2007 Chinese relation extraction task, a rather simple feature-based approach is able to outperform the best adopted parse tree kernel-based approach.

In this paper, we present a novel feature-based Chinese relation extraction framework, in which all the features do not require the Chinese word segmentation or deep natural language processing. Particularly, this framework is based on a 9-position structure feature between two entities. The design of this feature is motivated by the fact that there are some obvious connections between relation types/subtypes and position structures of two entities. For example, in many “Part-Whole” relation instances, one entity is often nested in the other entity, where *nested* is a position structure and Part-Whole is a relation type. In addition, compared with the 3-position structure implicitly or explicitly used in many feature-based methods, e.g., those in [Zhou et al. 2005; Che et al. 2005b; Chen et al. 2010], this 9-position structure is more discriminative since it is more effective to relieve the class imbalance problem. It is important to deal with this problem since there are far more negative relation instances than positive ones [Kambhatla 2006] and consequently this problem often hurts the performance of standard classifiers [Chawla et al. 2004].

In our framework, instead of trying to explore every feature reported in the literature [Zhou et al. 2005, 2007b; Che et al. 2005b; Jiang and Zhai 2007; Chen et al. 2010], our focus is to investigate the usefulness of our 9-positition structure. Therefore, we only complement the position structure feature with some basic character-based features, such as entity context (both internal and external) character *N*-grams and four word lists extracted from a published Chinese dictionary. After the classification with standard classifiers, we also derive some correction and inference mechanisms, in order to further improve the classified results. Specifically, at first, we rectify the classified relation types/subtypes by certain constraints, which are derived from the possible relation types/subtypes between any two entity types. Second, based on the relation hierarchy, a consistency check is carried out to make sure the relation type and the corresponding relation subtype are consistent. The aforesaid possible relations and relation hierarchy are available in the ACE task guideline. In addition to the above correction strategies, the entity co-reference information and some linguistic indicators are introduced to infer more positive relation instances through their links to the classified positive ones. It should be noted that this process can further integrate our strategies into a unified framework. Specifically, the classified results of different position structures can be linked together through the inferring process.

Experiments on the ACE 2005 data set show that the 9-position structure can provide strong support for Chinese relation extraction. Meanwhile, it can be captured with less effort than applying deep natural language processing. The entity co-reference does not help as much as we have expected. The lack of necessary annotations for the co-referenced entity mentions within a single document might be the main reason. By contrast, other strategies in our framework can further boost the extraction performance.

The remainder of this paper is organized as follows. Section 2 briefly introduces the definition of the ACE relation extraction task and reviews the related work. Section 3 defines three types of features, namely position structure (including 9-position and 3-position), entity type and character-based features. Our feature-based Chinese relation extraction framework is proposed in Section 4. Experimental studies on the ACE 2005 Chinese data set are presented in Section 5. Finally, Section 6 concludes the paper.

2. BACKGROUND

2.1 Task Definition

The research on relation extraction has been initiated and promoted by the Message Understanding Conferences (MUCs) (MUC, 1987-1998) and the NIST Automatic Content Extraction (ACE) program[1] (ACE, 2001-2008). According to the ACE 2005 program[2], there are five primary ACE tasks, i.e., the detection and recognition of entities, values, temporal expressions, relations, and events. In this paper, we focus on the ACE Relation Detection and Recognition (RDR) task and directly use the available entity information. 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 or entity mentions[3]. For example, the sentence "George Bush traveled to France on Thursday for a summit" conveys the ACE-style relation "Physical.Located" between the entity mentions "George Bush" and "France", where "Physical" and "Located" are pre-defined relation type and subtype, respectively. "George Bush" is the Arg-1 and "France" is the Arg-2. We can say that "George Bush" is "Located" in "France", but not vice versa.

The task of relation extraction can be regarded as the problem to classify the relation type, relation subtype and the argument order of each relation instance between any two entity mentions. Formally, let $r = (s, em_1, em_2)$ denote a relation instance, where s is a sentence, em_1 and em_2 are two entity mentions in s, and em_1 either precedes or embeds em_2 in the text. Given all relation instances $\{r_i\}$, our goal is to learn a function that maps each relation instance r_i to a type $t \in T$ and a subtype $st \in ST$, and to identify the role (i.e., argument order Arg-1 or Arg-2) of the two entity mentions. Here, T denotes the set of pre-defined relation types plus the type *None*, and *ST* is the set of pre-defined relation subtypes plus the *None* subtype. *None* means that there is no relation between two entity mentions, or the relation is not annotated. The classified relation is correct if and only if its type/subtype is correct and its two arguments are in the correct order.

---

[1] http://projects.ldc.upenn.edu/ace/

[2] http://www.itl.nist.gov/iad/mig//tests/ace/2005/doc/ace05-evalplan.v2a.pdf

[3] Each entity may be mentioned more than once, and thus has several entity mentions in a document (see Fig. 2). In this paper, we consider the relation instances between two entity mentions which belong to different entities.

2.2 Related Work

The research on relation extraction can be roughly divided into two directions, i.e., feature-based and kernel-based. We first review the related work according to different directions, and then review the work particularly for Chinese language.

Feature-based approaches transform the context of two entities into a linear vector of carefully selected linguistic features based on different levels of text analysis, ranging from morphological analysis and part-of-speech (POS) tagging to full parsing and dependency parsing. Miller et al. [2000] augmented syntactic full parse trees with semantic information corresponding to entities and relations and built generative models for the augmented trees. Kambhatla [2004] employed maximum entropy (ME) models to combine diverse lexical, syntactic and semantic features derived from word, entity type, mention type, overlap, dependency and parse tree. Besides these features, Zhou et al. [2005] further explored other features derived from the base phrase chunking information, semi-automatically collected country name list and personal relative trigger word list; and then took into account all the features into the classification step, where Support Vector Machines (SVMs) [Joachims 1998] were selected as the classifiers. 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. The reason could be that if combining several feature subspaces into one subspace, different original subspaces might have too much overlaps [Zhou et al. 2009a]. To avoid such a feature overlapping problem, Zhou et al. [2009a] proposed a multi-view approach to relation extraction.

On the other hand, kernel-based approaches design kernel functions on the relation context's structured representation such as parse tree or dependency tree, and then compute the similarity between two relation instances. 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 the similarity of 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. These three tree kernels require the matchable nodes to be at the same layer counting from the root and to have an identical path of ascending node from the roots to the current nodes, making their kernels with high precision but very low recall. Later, in order to incorporate the advantages of feature-based methods, Zhang et al. [2006] developed a composite kernel that combined convolution parse tree kernel with an entity kernel, and showed its effectiveness in capturing various syntactic features. Zhou et al. [2007a] experimented with a context-sensitive kernel by automatically determining context-sensitive tree spans, and applied a composite kernel to combine a convolution parse tree kernel and a state-of-art linear kernel for integrating both structured and flat features. Miyao et al. [2008] evaluated the usefulness of different syntactic parsers for the relation extraction carried out by SVMs with tree-kernels. Zhou et al. [2010] further integrated more syntactic and semantic information into the above context-sensitive convolution kernel. Katrenko et al. [2010] introduced local alignment kernels and explore various possibilities of using them for the relation extraction.

Besides the above supervised methods, some unsupervised methods [Takaaki et al. 2004; Chen et al. 2006a; Nakov and Hearst 2008] and semi-supervised methods [Zhang 2004; Chen et al. 2006b; Zhou et al. 2009b] were also explored. Unsupervised methods could overcome some difficulties in supervised approaches, such as labor intensive annotation efforts. However, they could hardly be directly applied in many NLP tasks since there is no relation type label attached to each instance in the clustering results [Chen et al. 2006b]. Therefore, semi-supervised methods have drawn much attention recently [Chen et al. 2006b].

The aforementioned works are mainly focused on English relations. Although Chinese processing is of the same importance as English and other Western language processing, unfortunately less work has been published on Chinese relation extraction. Che et al. (2005a) defined an improved edit distance kernel over the original Chinese string representation around particular entities. They studied only one ACE-style relation type, i.e., PERSON-AFFILIATION. Che et al. (2005b) explored several features and evaluated their performance on the ACE 2004 Chinese evaluation data. Huang et al. (2008) provided evidence showing that in ACE 2007 Chinese relation extraction, a rather simple feature-based approach is able to achieve reasonable performance (i.e. 0.63 F-measure); however, the best reported results of parse tree kernel-based approaches is unexpectedly low (i.e. 0.35 F-measure only). More recently, Zhang et al. (2009) proposed a composite kernel-based approach for ACE 2005 Chinese RDR task. Chen et al. (2010) adopted Deep Belief Network (DBN), and showed its effectiveness.

The insufficient study in Chinese relation extraction drives us to investigate how to find an approach that is particularly appropriate for Chinese. In this paper, we propose a novel position structure based framework for Chinese relation extraction. The contributions are three-fold. First, we propose a 9-position structure feature, which is used as the major component to form our framework. Second, we derive certain constraints based on possible relations and relation hierarchies, in order to improve the correctness and consistency of the classified relation types, subtypes and argument orders. Third, the entity co-reference information is used to infer more positive relation instances through their links to the classified positive ones.

3. FEATURE DESIGN

In this section, we describe the features used in our framework. In Table I, we first show the hierarchy of relation types and subtypes, as well as the frequencies of annotated (positive) relation instances on the ACE 2005 Chinese corpus.

Recall that our task is to identify the relations between any two entity mentions. Therefore, all the features are related to the entity mention pairs and their contexts. Specifically, for each pair of mentions, three kinds of features, namely position structure feature, entity type/subtype feature and character-based feature, are involved. For vector representations of features for the classification, please refer to Appendix B.

3.1 Position Structure Feature

Intuitively, the position structure of two entity mentions (em_1 and em_2) has some obvious connections with the type/subtype of the relation they might be. This can be understood from the following observations. In a lot of “Part-Whole” relation instances, the position structure of em_1 and em_2 tends to be nested. For example, in the sentence “The U.S.

Table I. The relation type/subtype hierarchy and the frequencies of annotated (positive) relation instances on the ACE 2005 Chinese corpus

Relation Type	Relation Subtype	Frequency
ART (artifact)	User-Owner-Inventor-Manufacturer	630
GEN-AFF (Gen-affiliation)	Citizen-Resident-Religion-Ethnicity	746
	Org-Location	1191
ORG-AFF (Org-affiliation)	Employment	1584
	Founder	17
	Ownership	25
	Student-Alum	72
	Sports-Affiliation	69
	Investor-Shareholder	85
	Membership	346
PART-WHOLE (part-whole)	Artifact	14
	Geographical	1289
	Subsidiary	983
PER-SOC (person-social)	Business	188
	Family	384
	Lasting-Personal	88
PHYS (Physical)	Located	1358
	Near	230

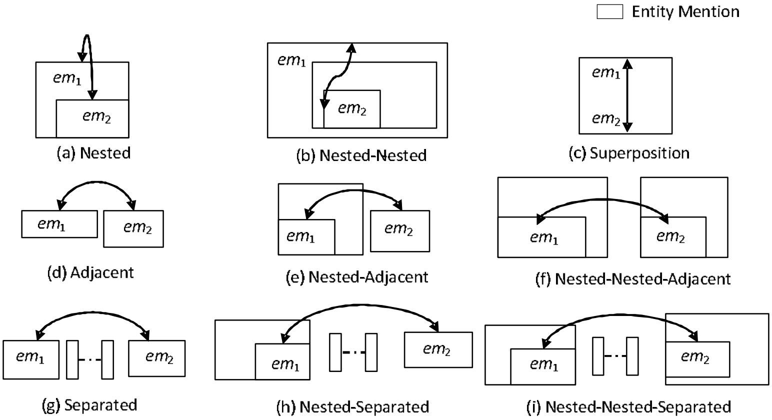

Entity Mention
em1
em2
(a) Nested
(b) Nested-Nested
(c) Superposition
(d) Adjacent
(e) Nested-Adjacent
(f) Nested-Nested-Adjacent
(g) Separated
(h) Nested-Separated
(i) Nested-Nested-Separated

Fig. 1. Nine position structure types, where each box is an entity mention.

Congress decided to veto the ecology bill”, the two nested mentions, em_1 (“The U.S. Congress”) and em_2 (“U.S.”) have a “Part-Whole.Subsidiary” relation. In addition, for many “Physical.Located” relations, the position structure of em_1 and em_2 is more likely to be adjacent, i.e., em_1 and em_2 are not nested and there is no entity mention in between them. For example, in a sentence “thousands of Palestinians rushed the Israeli checkpoint”, the relation of the two adjacent mentions, em_1 (“thousands of Palestinians”) and em_2 (“the Israeli checkpoint”), is “Physical.Located”. These observations drive us to analyze the position structure of the two entity mentions in-depth. We define nine types of the position structure as illustrated in Fig. 1. The formal definition for these 9-position structure types is given in Appendix A. Appendix C presents one Chinese example (selected from the ACE 2005 dataset for Chinese relation extraction) for each position structure. Here, we briefly explain these nine position structures.

For the structure types (a), (b) and (c), em_2 is nested (i.e. included) in em_1. In (a), there are no other entity mention that includes em_2 and is also nested in em_1. In (b), there is at least one entity mention (not em_1 or em_2) that includes em_2 and is also nested in em_1. In (c), em_1 includes em_2, and em_2 includes em_1 as well.

For the structure types (d), (e) and (f), em_1 and em_2 are not nested and there are no other full entity mentions in between them, even though there could be some characters in between em_1 and em_2. In (d), neither of the two entity mentions is nested in other entity mentions. In (e), em_1 or em_2 is nested in another entity mention. In (f), both em_1 and em_2 are nested in other entity mentions.

For the structure types (g), (h) and (i), em_1 and em_2 are not nested and there is at least one full entity mention in between them. In (g), neither of the two entity mentions is nested in other entity mentions. In (h), em_1 or em_2 is nested in other entity mentions. In (i), both em_1 and em_2 are nested in other entity mentions.

On the other hand, we can merge structure types (a) (b) and (c) into one single structure type. Similarly, we can merge the structure types (d) (e) and (f), as well as combine the types (g) (h) and (i). This means that one can combine structures of each row in Fig. 1 into one logical structure with a logical “or”. As a result, we can obtain three position structures, i.e., Nested+, Adjacent+, Separated+, each corresponding to one row in Fig. 1. This 3-position structure feature has been explicitly or implicitly adopted in several methods, e.g., in [Zhou et al. 2005; Che et al. 2005b; Jiang and Zhai 2007; Chen et al. 2010]. Specifically, this 3-position structure feature is exactly the position structure feature in [Chen et al. 2010]. Zhou et al. [2005] defined an Overlap category of features, which consider if one entity mention is included (or called nested) in the other entity mention, and if there are words or other entity mentions in between the two concerned entity mentions. Che et al. [2005b] adopted an Order feature, which also considers if one entity mention is included in the other one. We also think that in the parse tree feature spaces, e.g., those in [Jiang and Zhai 2007], the position structures of two entity mentions are implicitly considered.

3.1.1 Class Imbalance Problem

We analyze the difference between the 9-position structure feature and the 3-position one in terms of the effectiveness in solving the class imbalance problem. This problem typically occurs when there are far more instances of some classes than those of others. In such cases, standard classifiers tend to be overwhelmed by large classes and ignore the small ones and consequently cause a significant bottleneck in performance [Chawla et al. 2004]. The task of relation extraction encounters the class imbalance problem [Culotta et al. 2006; Kambhatla 2006], i.e., there are much more *None* (negative) class relation instances than ACE annotated (positive) class relation instances. For instance, in Tables II and III, the overall ratio of positive to negative class is 1:12.01 on ACE 2005 corpus.

If we divide all the relation instances according to different position structure types, we can observe that: compared with the situation of the 3-positon structures, the class imbalance problem with respect to the 9-positon structures is less serious for the majority (>99%) of relation instances. Specifically, in 3-position structures, the ratios of positive to negative relation instances for Nested+, Adjacent+ and Separated+ are 1:0.7273, 1:13.3629 and 1:85.1853, respectively. On the other hand, in 9-position structures, the ratios for Nested, Adjacent and Separated are 1:0.37, 1:6.82, and 1:42.87, respectively. This relieves the class imbalance problem a lot. It should be noted that these main structures, i.e., Nested, Adjacent, and Separated, occupy most (>99%) of the positive relation instances. Especially, the Nested structure is the most important one since it has about 68% of all the positive relation instances. We can see that the positive-to-negative ratio of Nested structure is much larger than the overall ratio.

Table II. The ratios of positive to negative relation instances on 3-position structures

Structure types	**#Positive class**	**#Negative class**	**Ratio**
Nested+	6332	4612	1 : 0.7283
Adjacent+	2028	27100	1 : 13.3629
Separated+	939	79989	1 : 85.1853
Overall	9299	111701	1 : 12.01

Table III. The ratios of positive to negative relation instances on 9-position structures

Structure Types	**#Positive Class**	**#Negative Class**	**Ratio**
Nested	6325	2347	1 : 0.37
Adjacent	1978	13501	1 : 6.82
Separated	928	39808	1 : 42.87
Superposition	6	407	1 : 67.84
Nested-Nested-Adjacent	50	3480	1 : 69.60
Nested-Nested-Separated	10	9142	1 : 914.20
Nested-Nested	1	1858	1 : 1858.00
Nested-Adjacent	0	10119	1 : INF
Nested-Separated	1	31039	1 : 31039.00
Overall	9299	111701	1 : 12.01

3.2 Entity Type and Subtype Features

These two features are concerned with the entity type and subtype of both entity mentions (i.e., em_1 and em_2). Entity mentions inherit the attributes (i.e. entity type and subtype) from the corresponding entity. Fig. 2 shows the dependency between entity and its mentions. For each mention pair, the combination of their entity types is for entity type feature, and similarly their entity subtypes are for the entity subtype feature.

The ACE 2005 categorizes entities into seven types (see Table IV), including "PER", "ORG", "GPE", "LOC", "FAC", "WEA" and "VEH". Each type is further divided into subtypes (see Table IV).

Table IV. The entity type/subtype hierarchy on ACE 2005 Chinese corpus

Entity Type	**Entity Subtypes**
PER (Person)	Group, Indeterminate, Individual
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
GPE (Geo-Political)	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
FAC (Facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
WEA (Weapon)	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified
VEH (Vehicle)	Air, Land, Subarea-Vehicle, Underspecified, Water

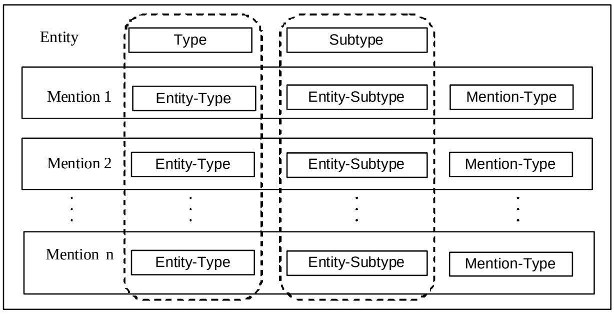

Entity
Type
Subtype
Mention 1
Entity-Type
Entity-Subtype
Mention-Type
Mention 2
Entity-Type
Entity-Subtype
Mention-Type
Mention n
Entity-Type
Entity-Subtype
Mention-Type

Fig. 2. The dependency between entity and its mentions.

3.3 Character-based Features

Character-based features involve *N*-gram features and wordlist-based features. Before describing them, we extract three types of character sequences from the context where two entity mentions appear. Note that we use characters instead of words.

3.3.1 Character Sequences

- Internal Character Sequence

These character sequences are concerned about the extents and the heads[4] of both entity mentions, and can be categorized into four types of sequences as follows:

Label	Scope
CME1	all the characters in em_1
CMH1	all the characters in the head of em_1
CME2	all the characters in em_2
CMH2	all the characters in the head of em_2

- In-Between Character Sequence

If em_1 (em_2) does not contain em_2 (em_1), then all the characters between two entity mentions will be extracted as the in-between character sequence.

- External Context Character Sequence

These character sequences are concerned with the characters around two entity mentions in a given window size w_s, and can be classified as the following four types:

Label	Scope
CBM1	at most w_s characters before em_1
CAM1	at most w_s characters after em_1
CBM2	at most w_s characters before em_2
CAM2	at most w_s characters after em_2

The extraction of external character sequences must comply with one rule, i.e., the extracted character sequence cannot enter into or move across any entity mentions.

3.3.2 Features from Character Sequences

- *N*-gram Features

All character sequences are then transformed into *N*-gram features. For example, supposing an extracted character sequence is $c_1c_2c_3c_4$, the Uni-gram feature is { c_1, c_2, c_3, c_4 }, and the Bi-gram feature is { c_1c_2, c_2c_3, c_3c_4 }. Each involved character sequence will be used to construct one Uni-gram feature as well as one Bi-gram feature.

Character Uni-gram features	**Character Bi-gram features**
CME1_Uni, CMH1_Uni, CME2_Uni, CMH2_Uni, In_Between_Uni, CBM1_Uni, CAM1_Uni, CBM2_Uni, CAM2_Uni	CME1_Bi, CMH1_Bi, CME2_Bi, CMH2_Bi, In_Between_Bi, CBM1_Bi, CAM1_Bi, CBM2_Bi, CAM2_Bi

---

[4] In ACE, each entity mention has a head annotation and an extent annotation, and the head word is usually more important than the other parts [Zhou et al. 2005; Li et al. 2007].

- Wordlist-based Features

With insufficient training data, many discriminative words for the relation extraction might not be covered by *N*-gram features. Therefore, we build wordlist-based features which are extracted from a published Chinese dictionary. These wordlists include Chinese preposition list (165 words), orientation list (105 words), auxiliary list (20 words), and conjunction list (25 words). It can be expected that some words in these wordlist can serve as strong indicators for some relation types or subtypes. For instance, if there is an orientation word “south” in the context of two entity mentions, it is more likely that these two mentions have a “Physical.Located” relation. The in-between and external context character sequences are transformed to wordlist-based features. On the other hand, the internal character sequences are not involved since they are not likely to include those words related to preposition, orientation, auxiliary or conjunction words. Each involved character sequence is used to construct one wordlist-based feature for every wordlist. Features with respect to different wordlist are different from each other.

4. A POSITION STRUCTURE BASED FRAMEWORK

Our relation extraction framework is summarized in Model 1, which is based on the 9-position structure. In Step 1 we divide all the relation instances into nine parts according to the 9-position structures defined in Section 3.1, in order to solve the class imbalance problem. The detailed motivation of this divide strategy has been discussed in Section 3.3.1, and is also verified by the experiments in Section 5.

Model 1: Position Structure Based Relation Extraction Framework

Step 1:	According to the nine position structures, divide all the relation instances into nine sets. Then, execute step 2 to 5 on each set.
Step 2:	Initially perform the relation detection and recognition in a cascade manner by standard classifiers.
Step 3:	Based on the possible relation information, verify the classified relation type/subtype and the argument order of every relation instances.
Step 4:	Carry out the consistency check between the relation type and subtype based on the relation hierarchy.
Step 5:	Infer more positive relation instances from the classified [5] positive relation instances based on co-reference information and linguistic indicators.

In Step 2, we initially perform the RDR task by a cascade strategy, i.e., carrying out the relation detection and recognition separately. Specifically, we first classify every relation instance as positive or negative. Then, we classify each positive relation as one of the relation type/subtype. Both classifications are carried out by standard classifiers (i.e. SVMs). The cascade strategy is against the all-at-one strategy, i.e., carrying out relation detection and recognition at one time by classifiers. We do not adopt the all-at-once strategy because the number of positive relation instances in any one relation type/subtype is much smaller than the number of negative ones. On the other hand, the cascade strategy can relieve the class imbalance problem due to the fact that the number of all positive relation instances is much bigger than that of positive ones in any one relation type/subtype. Then, we will explain the strategies in other steps (i.e. Step 3 to Step 5).

---

[5] The term “classified” means the state after the previous step in our framework. It does not necessarily only mean the state after the classification by standard classifiers.

4.1 Possible Relation between Arg-1 and Arg-2

In many tasks of information extraction, such as entity extraction and relation extraction, some prior knowledge is usually involved that can be useful to the tasks. In ACE 2005 guidelines, a table (e.g. Table V) of possible relation between Arg-1 and Arg-2 is provided. Given two entity mentions, the possible relation type and subtype can be obtained according to the two entity types (listed in Table IV). For instance, according to Table V, if both entity types of Arg-1 and Arg-2 are PER (person), the possible relation type can be Per-Social, and the relation subtypes can be Business or Family. If the entity type of Arg-1 is PER and that of Arg-2 is ORG (organization), the possible relation type can be Org-Aff (Org-affiliation).

Table V. Examples of the possible relations between Arg-1 and Arg-2 [6]

	PER	ORG	GPE
PER	Per-Social.Bus. Per-Social.Family …	Org-Aff.Employment, Org-Aff.Ownership, Org-Aff.Student-Alum, Org-Aff.Sports-Affiliation …	Physical.Located, Physical.Near, Org-Aff.Employment …
ORG	…	Part-Whole.Subsidiary, Org-Aff.Investor-Shareholder …	Part-Whole.Subsidiary, Org-Aff.Investor-Shareholder …
GPE	…	Org-Aff.Investor-Shareholder, Org-Aff.Membership …	Physical.Near, Part-Whole.Geographical …

The first column and row represent the entity type of Arg-1 and that of Arg-2, respectively.

This kind of prior knowledge has two important roles. First, we can rectify the relation type/subtype classified by SVMs. According to the entity types of two entity mentions, if the classified relation type/subtype is not possible, then we will revise the type/subtype to *None*. Second, if the relation type/subtype is possible, we then adjust the argument order of the two entity mentions.

In many feature-based models [Kambhatla 2004; Zhou et al. 2005; Wang et al. 2006], they used a different approach to the argument order problem. Specifically, except for symmetric relations, the argument order is modeled by considering one relation subtype as two new relation subtypes with different orders. For example, the relation subtype Physical.Located is changed to two relation subtypes, namely em_1-Physical.Located-em_2 and em_2-Physical.Located-em_1, where the former denotes that em_1 is the Arg-1, and the latter denotes that em_2 is the Arg-1. There are two drawbacks of their strategy. First, it could be more time-consuming since it involves almost twice number of classifiers we need. Second, it may make the class imbalance problem more serious because the number of positive relation instances in each (new) relation subtype becomes smaller than the number of the positive instances in each (original) relation subtype.

---

[6] This table is only part of the original table in the ACE 2005 Chinese relation extraction guidelines (http://projects.ldc.upenn.edu/ace/annotation/2005Tasks.html)

4.2 Interactive Consistency Check Using Relation Hierarchy

In our framework, the relation type and subtype are classified separately and they may not be consistent. We then try to make them consistent according to the relation hierarchy (see Table I in Section 3). There are some existing strategies, such as strictly Bottom-Up [Kambhatla 2004; Zhou et al. 2005] and Guiding Top-Down, to deal with this problem. With regard to the strictly Bottom-Up strategy, the relation type should conform to the relation subtype. Once the subtype is recognized, the type is determined by the subtype, since a subtype belongs to one unique type. As for the Guiding Top-Down strategy, the upper level (relation type) guides the down-level (relation subtype). It assumes that the classification result of relation type is more precise. As a result, the subtype will be revised to *None* if it does not conform to the type.

However, we think that these two strategies lack necessary interaction between two levels, and hence do not make full use of both levels' classification results. Therefore, we derive the following consistency check strategy.

Procedure 1: Type Selection based Consistency Check

Input:	Classified pair (type, subtype)
Output:	Consistent pair (c-type, c-subtype)
Parameters:	*cn*
Step 1:	Select *cn* most likely types based on the probabilities of the classification results. For every candidate type, if it conforms to the subtype, then c-type:=this type, c-subtype:= subtype. Return.
Step 2:	If no candidate type conforms to the subtype, then c-type = *None*; c-subtype:=*None*; Return

Similarly, we can have the Subtype Selection based consistency check strategy, which selects *cn* most likely subtypes, and check them against the types.

4. 3 Inferring More Positive Relation Instances

The relation extraction performances of different position structures have great disparity. Our experiments show that the performances of the Nested and Adjacent structures are much better than the results of the other seven structures. In fact, there are almost no positive relation instances classified for the other seven position structures. This phenomenon may have two reasons. First, the imbalance class problems in the other seven position structures are much more serious, as evidenced in Table III. Second, intuitively, Nested and Adjacent relation instances are more likely to be positive classes (or more likely to be annotated) and hence can be extracted easily.

There are some linguistic homogeneous characteristics (such as co-reference) that can be used to infer more positive relations through the classified positive ones. Specifically, after obtaining one classified positive relation with Nested or Adjacent structure, we can assign its relation type/subtype to other relation instances with different position structure but sharing the same attributes. These attributes are related to co-reference information and pattern-based information, which will be described below.

4.3.1 Co-reference based Inference

Each entity mention belongs to only one entity, and hence naturally inherits the type and subtype attributes from the corresponding entity (see Fig. 2). Entity mentions are considered as co-referent when they belong to the same entity.

Once Nested and Adjacent relation instances are recognized as positive, the co-reference information can be adopted to carry out the relation inferring. Specifically, if a relation instance with different position structure has the same two entities as in the classified positive one, this relation instance will be classified as the same relation type and subtype. 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 $e_1=\{em_{11}, em_{12}, \ldots, em_{1n}\}$ and $e_2=\{ em_{21}, em_{22}, \ldots, em_{2m}\}$ (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.

When considering the co-reference information, we may find another type of inconsistency, e.g., $R(em_{11}, em_{21}) \neq R(em_{12}, em_{22})$, where (em_{11}, em_{21}) and (em_{12}, em_{22}) are different in their contexts or structures, and R denotes the classified relation type/subtype. The co-reference not only helps for inference but also provides a chance to check the consistency among entity mention pairs. As the classification results of SVM can be transformed to probability by a sigmoid function

$$P(R(r)=t\,|\,y_t)=\frac{1}{1+e^{-y_t}} \qquad (1)$$

, the relations of lower probability mention pairs can be revised according to the relation of highest probability mention pairs. In Eq. 1, the left side denotes that the probability of relation type/subtype t for relation instance r and y_t is the output value of the t by the classifiers.

4.3.2 Pattern-based Inference

The classified positive relation instances of adjacent structure can infer more relation instances of separated structure if there are some linguistic indicators in the local context. For example, given a local context “both em_1 and em_2 are located in em_3”, if em_2 and em_3 are classified as a positive relation instance, em_1 and em_2 will have the same relation type/subtype as that em_2 and em_3 hold. Currently, the indicators under consideration are “and” and “or”. However, more patterns can be included in the future.

5. EXPERIMENTAL EVALUATION

5.1 Evaluation Data Sets

We evaluate our relation extraction framework on the training dataset for the ACE 2005 Chinese Relation Detection and Recognition (RDR) [7] task provided by the Linguistic Data Consortium (LDC). The 633 documents have been manually annotated with 9299 instances of relations. Meanwhile, 6 relation types and 18 subtypes are pre-defined. More details are shown in Table I in Section 3. Because of no test data at hand, we randomly select 75% out of the 633 documents as the training data and the remaining documents are used for evaluation. All the reported performances in this paper on the ACE RDR 2005 corpus are evaluated using 4-fold cross validation on the entire corpus. In this paper, we only measure the performance of relation extraction model on "true" mentions with "true" chaining of co-reference (i.e., they are annotated by LDC annotators).

---

[7] http://www.itl.nist.gov/iad/mig//tests/ace/2005/doc/ace05-evalplan.v2a.pdf

5.2 Evaluation Set-up

The aim of our experiments is to evaluate the performance of the proposed features (especially the 9-position structure feature) in Section 3, as well as each step in our relation framework in Section 4. Two baseline methods are involved. Both of them carry out the RDR task as an all-at-once multi-class classification problem. In the first baseline method, the involved features are the 3-position structure feature in Section 3.1 and other features in Sections 3.2 and 3.3. Recall that (see Section 3.1) the 3-position feature is implicitly or explicitly used in many feature-based methods, e.g., those in [Zhou et al. 2005, 2007b; Che et al. 2005b, Chen et al. 2010]. In the second baseline method, the 9-position structure feature and other features are adopted. The first baseline (denoted as 3-Position Baseline) is used to test whether the 9-position feature is helpful, while the second baseline (denoted as 9-Position Baseline) is to test the performance of each step in our framework. When evaluating each step, its following steps will not be executed.

Besides the above main aims, we also evaluate the roles of different categories of features in Section 3 played in our framework. In addition, we provide a performance comparison between our framework and the kernel based framework in [Zhang et al. 2009], which also adopted the 9-position feature (slightly different from ours), and carried out the ACE 2005 Chinese RDR task as well. Finally, since the dimensions of the vector representations for all the features are very large, we would like to study the effectiveness of some feature selection methods such as Information Gain (IG) [Yang and Pedersen 1997] and Bi-norm Separation (BNS) [Forman 2003].

The SVMlight [Joachims 1998] with linear kernel and default configuration is adopted as the classification tool. In the step 1-3 of our framework, for every entity mention pair (em_1, em_2) in a sentence, we simply choose em_1 as Arg-1, and em_2 as Arg-2, where em_1 precedes or contains em_2.The window size (w_s) of character-based features is 4. The options count *cn* in the type and subtype selection based consistency check strategy (see Section 4.2) are all set to 2.

As for the evaluation metrics, we adopt three primary metrics, i.e., Precision, Recall, F-measure, which are also commonly used to evaluate other relation extraction methods, e.g., those in [Zhou et al. 2005, 2007a,2007b, 2010; Jiang and Zhai 2007; Zhang et al. 2006; Chen et al. 2010]. In addition, the Wilcoxon signed rank test is adopted as the measure of the statistical significance of the improvements over baseline methods. The improvements (at significance level 0.05) over the 3-Position Baseline and 9-Position Baseline are denoted as "α" and "β", respectively, in the result table. In each table, both the performance of positive relation types and those of positive relation subtypes will be reported. All results are the average ones over 4-fold experiments. Note that the results are slightly different from those in [Li et al. 2008], which did not involve the 4-fold experiments.

5.3 Evaluation on the 9-PositionStructure Feature

In this set of experiments, firstly, we will compare our 9-position feature with the 3-position feature, when all other features are involved and we do not divide the relation instances. Secondly, we evaluate our Divide strategy (Step 1) in our framework.

Table VI summarizes the experimental results. We can have the following conclusions. Firstly, when we do not divide relation instances, the 9-Position Baseline significantly outperforms 3-Position Baseline, which shows the effectiveness of our 9-position feature. This is due to that the class imbalance problem of the 3-positon is more serious than that of the 9-position (see Section 3.1 and Tables II and III). Secondly, the 9-Position-Divide significantly further improves the F-measure 18.15% and 23.15% over 9-Position Baseline in relation types and relation subtypes recognition, respectively.

Table VI. Evaluation of position structure feature

Types/Subtypes	Precision (%)	Recall (%)	F-measure (%)
3-Position Baseline	73.06/71.54	34.84/31.27	47.18/43.52
9-Position Baseline	72.65/72.51	45.21$^{\alpha}$/39.91$^{\alpha}$	55.73$^{\alpha}$/51.48$^{\alpha}$
9-Position_Divide	**77.39$^{\alpha\beta}$/75.00$^{\alpha}$**	**57.31$^{\alpha\beta}$/54.91$^{\alpha\beta}$**	**65.85$^{\alpha\beta}$/63.40$^{\alpha\beta}$**

5.4 Evaluation on the Cascade Strategy

The aim is to investigate the effectiveness of the cascade strategy, i.e., the step 2 in our framework. In the detection stage, binary-class SVMlight is adopted, while in the recognition stage, multi-class SVMlight is adopted. Table VII presents the experimental results. We can see that the Cascade strategy outperforms the all-at-once strategy.

Table VII. Evaluation of two detection and recognition modes

Types/Subtypes	Precision (%)	Recall (%)	F-measure (%)
All-at-once	**77.39$^{\alpha\beta}$/75.00$^{\alpha}$**	57.31$^{\alpha\beta}$/54.91$^{\alpha\beta}$	65.85$^{\alpha\beta}$/63.40$^{\alpha\beta}$
Cascade	74.48/71.99	**60.20$^{\alpha\beta}$/58.19$^{\alpha\beta}$**	**66.58$^{\alpha\beta}$/64.36$^{\alpha\beta}$**

5.5 Evaluation on the Role of Possible Relation Information

As discussed in Section 4.1, the possible relation information between Arg-1 and Arg-2 has two important roles: one is to rectify the classification results; the other is to adjust the argument order. Table VIII shows the performance of this step. We can clearly see that it is contributing and improves F-measure 2.82% and 5.26% in type and subtype recognition, respectively.

Table VIII. Evaluation of rectifying and adjusting based on the possible relations

Types/Subtypes	Precision (%)	Recall (%)	F-measure (%)
+Rectify +Adjust	76.58/77.48$^{\alpha}$	61.90$^{\alpha\beta}$/60.19$^{\alpha\beta}$	68.46$^{\alpha\beta}$/67.75$^{\alpha\beta}$

5.6 Evaluation on the Consistency Check Strategy

This is to test the step 4, i.e., the consistency check method in Section 4.2. Table IX shows the results, indicating that the strategies using subtypes to determine or select types (Type Selection) perform better than the Subtype Selection Strategy. This may be attributed to the fact that previous correction (in step 3) for relation subtype is better than that of relation type. Overall, the type selection based consistency check strategy is the best one.

Table IX. Comparison of different consistency check strategies

Types/Subtypes	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Guiding Top-Down	$77.48^{\alpha\beta}/77.91^{\alpha\beta}$	$61.88^{\alpha\beta}/58.83^{\alpha\beta}$	$68.81^{\alpha\beta}/67.04^{\alpha\beta}$
Subtype Selection	$79.47^{\alpha\beta}/77.52^{\alpha}$	$61.76^{\alpha\beta}/59.23^{\alpha\beta}$	$69.50^{\alpha\beta}/67.15^{\alpha\beta}$
Strictly Bottom-Up	$80.38^{\alpha\beta}/77.44^{\alpha}$	$\mathbf{62.45}^{\alpha\beta}/\mathbf{60.16}^{\alpha\beta}$	$70.29^{\alpha\beta}/67.71^{\alpha\beta}$
Type Selection	$\mathbf{80.81}^{\alpha\beta}/\mathbf{78.06}^{\alpha\beta}$	$62.31^{\alpha\beta}/60.04^{\alpha\beta}$	$\mathbf{70.36}^{\alpha\beta}/\mathbf{67.86}^{\alpha\beta}$

5.7 Evaluation on the Relation Inference

We first present the results (after Step 4) of Nested and Adjacent structures in Table X. The results of other structures are not shown since they are almost zero. According to these reported results and our discussion in Section 4.3, intuitively we can follow the path of "Nested ⇒ Adjacent ⇒ Separated ⇒ Others" to perform the inference. But soon we find that if two concerned entity mentions are nested, almost all the co-refereed mentions are nested. So basically inference works on the path "Adjacent ⇒ Separated⇒ Others''.

Table X. Evaluation results of different position structures after Step 4

Types/Subtypes	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Nested	80.47/77.41	85.39/82.15	82.86/79.71
Adjacent	85.81/84.50	19.87/19.57	32.27/31.77

Then, through this inference path, we use both co-reference information and linguistic indicators to construct relation inferring. The performance of relation inferring is summarized in Table XI. We can see that the inferring step does not help as much as we have expected. This might be due to that the lack of enough annotated relations for co-reference mentions and for those sharing the same patterns, i.e., linguistic indicators.

Table XI. Evaluation of the relation inference

Types/Subtypes	**Precision (%)**	**Recall (%)**	**F-measure (%)**
+Inference	$80.71^{\alpha\beta}/77.75^{\alpha}$	$62.48^{\alpha\beta}/60.20^{\alpha\beta}$	$70.43^{\alpha\beta}/67.86^{\alpha\beta}$

5.8 Evaluation on the Role of Every Feature Category

Then, we evaluate the contribution of every feature category for our framework. All the steps in our framework (see Section 4) are involved, but we will adopt the features incrementally. Only entity type and subtype features do not work. Therefore, Table XII shows the results when we incrementally add the 9-position structure, the external contexts and internal contexts, Uni-grams and Bi-grams, and at last the word lists on them. The observations are: first, the 9-position structure provides stronger support than other individual features. Second, Uni-grams provide more discriminative information than Bi-grams. Third, external context seems more useful than internal context. At last, the wordlist feature slightly improves the performance.

Table XII. Evaluation of the feature design

Types/Subtypes	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Entity Type + Position Structure	71.38/67.23	$50.51^{\alpha\beta}$/$47.58^{\alpha\beta}$	$59.16^{\alpha\beta}$/$55.72^{\alpha\beta}$
+ External (Uni-)	$77.53^{\alpha\beta}$/72.85	$59.39^{\alpha\beta}$/$55.81^{\alpha\beta}$	$67.26^{\alpha\beta}$/$63.20^{\alpha\beta}$
+ Internal (Uni-)	$80.06^{\alpha\beta}$/76.75^{α}	$62.17^{\alpha\beta}$/$59.60^{\alpha\beta}$	$69.99^{\alpha\beta}$/$67.09^{\alpha\beta}$
+ Bi- (Internal and External)	$80.64^{\alpha\beta}$/77.58^{α}	**62.54**$^{\alpha\beta}$/$60.17^{\alpha\beta}$	**70.45**$^{\alpha\beta}$/$67.77^{\alpha\beta}$
+ Wordlist	**80.71**$^{\alpha\beta}$/**77.75**$^{\alpha}$	$62.48^{\alpha\beta}$/**60.20**$^{\alpha\beta}$	$70.43^{\alpha\beta}$/**67.86**$^{\alpha\beta}$

5. 9 Comparison with the Kernel-based Approach

We provide a performance comparison between our framework and the kernel based framework in [Zhang et al. 2009], which was also evaluated on the ACE 2005 Chinese RDR task (for relation type only). Table XIII reports the results, which shows that our approach outperforms this kernel-based approach, although it uses features that are similar to those in our framework.

Table XIII. Comparison with kernel-based approach

Types	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Kernel-based	**81.83**	49.78	61.90
Ours	80.71	**62.48**	**70.43**

5.10 Studies on Feature Selection Methods

Since the large dimension and serious sparseness of the vector representation, we would like to test whether the feature selection methods can be useful in our task. Two feature selection methods (IG and BNS) are investigated.

Information gain [Yang and Pedersen 1997] of a term measures the number of bits of information obtained for category prediction by the presence or absence of the term in a document. Let *m* be the number of classes. The information gain of a term *t* is defined as:

$$IG(t) = -\sum_{i=1}^{m} p(c_i)\log p(c_i) + p(t)\sum_{i=1}^{m} p(c_i \mid t)\log p(c_i \mid t) + p(\bar{t})\sum_{i=1}^{m} p(c_i \mid \bar{t})\log p(c_i \mid \bar{t})$$

Forman [2003] presented an empirical comparison of twelve feature selection methods. Results revealed the surprising performance of a new feature selection metric, “Bi-Normal Separation” (BNS).Let *tp*(*t*) be true positives (number of positive cases containing term *t*), *fp*(*t*) be false positives (number of negative cases containing term *t*), *pos* denote the number of positive cases, *neg* be the number of negative cases, *tpr*(*t*) denote the sample true positive rate (*tp*(*t*)/*pos*) and *fpr*(*t*) be the sample false positive rate (*fpr*(*t*)/*neg*). BNS can be defined as follows:

$$BNS(t) = |\mathrm{F}^{-1}(tpr(t)) - \mathrm{F}^{-1}(fpr(t))|,\ \ where\ \ \mathrm{F}\ \ is\ \ the\ \ Normal\ \ c.d.f$$

For each method mentioned above, we implement feature selection in two ways. One is to construct feature selection on the whole feature space. The other is to implement feature selection on N-gram sub-features, e.g., left-4 context Uni-gram, while holding entity type/subtype and wordlist-based features unchanged. The latter strategy gain better performance according to the experimental results in Fig. 3 and 4, where “Previous” corresponds to the result without feature selection. Although fewer features can reduce the time cost of classifiers, the relation extraction results do not seem to be promising. This might be because that SVM itself has enough power to find the discriminative dimensions on the given data set. To continue this direction, maybe some other formal methods, like the PLSI [Hofmann 1999] which has successful application in text processing. This remains as our future work.

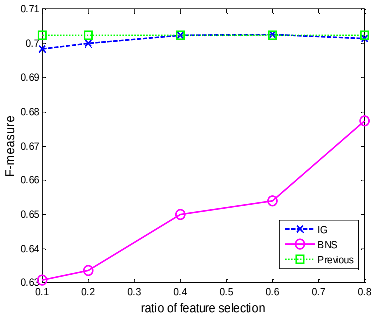


Fig. 3. Feature selection on whole subspace

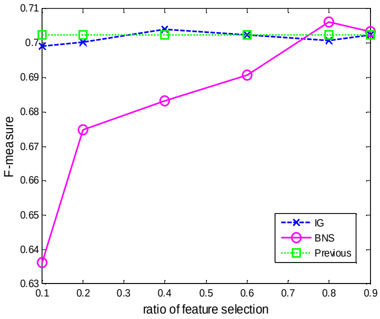


Fig. 4. Feature selection on *N*-gram sub-features

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a position structure based framework for Chinese entity relation extraction. The main contributions can be concluded as follows. First, a 9-position structure feature, which is conceptually clear and computationally efficient, is devised to relieve the serious class imbalance problem. This feature is also used as a major component to form our divide-and-conquer relation extraction framework. Second, the possible relation based constraints are used to verify the relation classification results and adjust the argument orders of relations. Third, an interactive consistency checking strategy is proposed to check whether the classified type and subtype conform to the given relation hierarchy. Last but not the least, co-reference information and pattern-based features are used to infer the more positive relations through the classified positive ones. The effectiveness of them, especially the position structure feature, has been demonstrated in the experiments conducted on the ACE 2005 Chinese data set.

Although the inferring step has not received the convincing performance improvement, this direction could be interesting and fruitful. It is because that the inferring can be derived from a graph where a vertex represents an entity, and the initial edge is the classified relations of Nested and Adjacent structure. Then, the other relations of any structure could be represented by this graph. Moreover, this graph can represent the relations of entity pairs which are not in one sentence. We will investigate this direction in the future. Furthermore, as for the efficiency issue of the proposed framework, we would like to investigate the usefulness of *l*1-norm SVM, which it is efficient in dealing with large-scale data sets [Sra, 2006]. We would also systematically investigate its effectiveness in improving the performance of the relation extraction.

APPENDIX A: FORMAL DEFINITION FOR THE 9-POSITION STUCTRURE

Given an entity mention *em*, let *em.start* and *em.end* denote the start and end positions of *em* in a sentence respectively. Let $em_i \supset em_j$ denote $(em_i.start, em_i.end) \supset (em_j.start, em_j.end)$ and $(em_i.start, em_i.end) \neq (em_j.start, em_j.end)$, and let $em_k \perp (em_1, em_2)$ denote $em_1.end < em_k.start$ and $em_k.end < em_2.start$. For any two entity mentions em_1 and em_2, where $em_1 \supset em_2$ or em_1 precedes em_2, the position structure of them can be grouped into nine categories in Table XIV.

Table XIV. Formal Definition for the 9-Position structures

Type	**Condition**	**Label** *
Nested	$em_1 \supset em_2 \wedge \neg\exists (em_i)(em_1 \supset em_i \wedge em_i \supset em_2)$	(a)
Nested-Nested	$em_1 \supset em_2 \wedge \exists (em_i)(em_1 \supset em_i \wedge em_i \supset em_2)$	(b)
Superposition	$em_1.start = em_2.start$ and $em_1.end = em_2.end$	(c)
Adjacent	$em_1.end < em_2.start \wedge \neg\exists (em_i)(em_i \supset em_1 \vee em_i \supset em_2) \wedge \neg\exists (em_j)(em_j \perp (em_1, em_2))$	(d)
Nested-Adjacent	$em_1.end < em_2.start \wedge (\exists (em_i)(em_i \supset em_1 \wedge \neg\exists (em_j)(em_j \supset em_2)) \vee \exists (em_i)(em_i \supset em_2 \wedge \neg\exists (em_j)(em_j \supset em_1))) \wedge \neg\exists (em_j)(em_j \perp (em_1, em_2))$	(e)
Nested-Nested-Adjacent	$em_1.end < em_2.start \wedge \exists (em_i)(em_i \supset em_1) \wedge \exists (em_j)(em_j \supset em_2) \wedge \neg\exists (em_j)(em_j \perp (em_1, em_2))$	(f)
Separated	$\exists (em_j)(em_j \perp (em_1, em_2)) \wedge \neg\exists (em_i)(em_i \supset em_1 \vee em_i \supset em_2)$	(g)
Nested-Separated	$\exists (em_j)(em_j \perp (em_1, em_2))(\exists (em_i)(em_i \supset em_1 \wedge \neg\exists (em_j)(em_j \supset em_2)) \vee \exists (em_i)(em_i \supset em_2 \wedge \neg\exists (em_j)(em_j \supset em_1)))$	(h)
Nested-Nested-Separated	$\exists (em_j)(em_j \perp (em_1, em_2)) \wedge \exists (em_i)(em_i \supset em_1) \wedge \exists (em_j)(em_j \supset em_2)$	(i)

* Corresponding examples are illustrated in Fig. 1

APPENDIX B: FEATURE REPRESENTATION FOR CLASSIFICATION

Once the features are obtained, the task of the Chinese Relation Extraction is modeled as a multi-class classification problem. Support Vector Machine (SVM) [Boser et al. 1992; Cortes and Vapnik 1995] is selected as the classification tool since it represents the state-of-the-art in the machine learning research. Given a training set of labeled instance pairs $(\mathrm{x}_i, y_i), i = 1, \ldots, l$ where $\mathrm{x}_i \in \mathrm{R}^n$ and $\mathbf{y}_i \in \{1, -1\}^l$, SVM requires the solution of the following optimization problem:

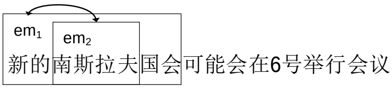
$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b) \geq 1-\xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

We use SVM in both the relation detection process and relation type and subtype recognition process. 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 use binary vector representation for the features obtained before, as explained in Table XV. We then combine the following vectors into a single feature vector to SVM.

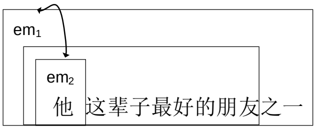
Table XV. Feature vector representation

Feature	**Representation**
Position Structure	One 9-dimensional binary vector where the i^{th} entry is 1 if the position structure is of the i^{th} type, and the other entries are 0
Entity Type and Subtype	For each entity mention pair, one binary vector for entity type and one binary vector for entity subtype where the dimensions of them are the total numbers of the entity types and the subtypes ACE defines and the i^{th} entry of the corresponding vector is 1 if the i^{th} type or subtype is recognized.
N-gram	For each internal and external context character string (sequence), one binary vector for Uni-grams and one binary vector for Bi-grams, where the dimensions of them are the total numbers of Uni-grams and Bi-grams in the whole corpus respectively and the i^{th} entry of the corresponding vector is 1 if the i^{th} Uni-gram or Bi-gram appears in the given character sequence.
Wordlist	For each in-between and external context character string, one 4-dimensional vector, where each entry corresponds to one wordlist and the i^{th} entry is 1 if the corresponding string contains any word in the i^{th} wordlist.

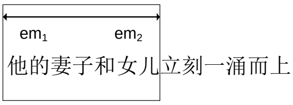
APPENDIX C: CHINESE EXAMPLES FOR 9-POSITION STRUCTURES


em$_1$
em$_2$
新的南斯拉夫国会可能会在6号举行会议

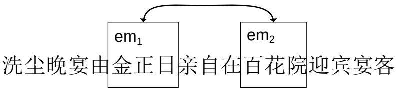
(a) Nested Structure with Relation Type/Subtype: PART-WHOLE/ Subsidiary


em$_1$
em$_2$
他这辈子最好的朋友之一

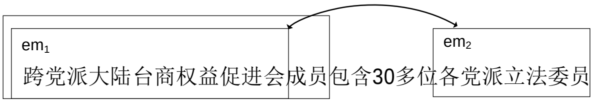
(b) Nested-Nested Structure with Relation Type/Subtype: PER-SOC/Lasting-Personal


em$_1$
em$_2$
他的妻子和女儿立刻一涌而上

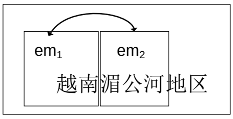
(c) Superposition Structure with Relation Type/Subtype: PER-SOC/Family


em$_1$
em$_2$
洗尘晚宴由金正日亲自在百花院迎宾宴客

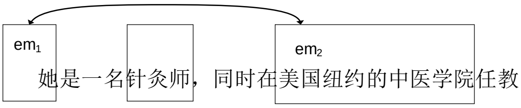
(d) Adjacent Structure with Relation Type/Subtype: PHYS/Located


em$_1$
em$_2$
跨党派大陆台商权益促进会成员包含30多位各党派立法委员

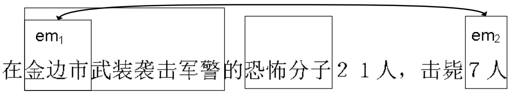
(e) Nested-Adjacent Structure with Relation Type/Subtype: None/None


em$_1$
em$_2$
越南湄公河地区

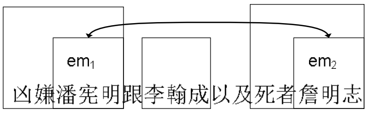
(f) Nested-Nested Adjacent Structure with Relation Type/Subtype: PART-WHOLE/Geographical


em$_1$
em$_2$
她是一名针灸师，同时在美国纽约的中医学院任教

(g) Separated Structure with Relation Type/Subtype: ORG-AFF/Employment


em$_1$
em$_2$
在金边市武装袭击军警的恐怖分子２１人，击毙７人

(h) Nested-Separated Structure with Relation Type/Subtype: PHYS/Located


em$_1$
em$_2$
凶嫌潘宪明跟李翰成以及死者詹明志

(i) Nested-Nested-Separated Structure with Relation Type/Subtype: PER-SOC/Business

ACKNOWLEDGMENTS

We would like to thank the editor and reviewers for their constructive comments. This work was supported in part by the Hong Kong RGC (Project Number: CERG PolyU5211/05E), China's NSFC (Grant No: 61070044), the Basic Application Research Project of Tianjin, China (Grant No: 09JCYBJC00200), and one NSFC-RSE joint project.

REFERENCES

BOSER B. E., GUYON I., AND VAPNIK V. 1992. A training algorithm for optimal margin classifers. *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory,* pp. 144-152. ACM Press

BUNESCU R. AND MOONEY R. 2005. A Shortest Path Dependency Tree Kernel for Relation Extraction, *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pages 724-731.

CHAWLA N., JAPKOWICZ N. AND KOLCZ A. 2004. Editorial: Special Issue on Learning from Imbalanced Datasets. *SIGKDD Explorations*, vol. 6, no. 1, pp. 1-6, 2004.

CHE W. , JIANG J., SU Z., PAN Y. AND LIU T. 2005a. Improved-Edit-Distance Kernel for Chinese Relation Extraction. *In Proceedings of The Second International Joint Conference on Natural Language Processing (IJCNLP-05)*, 2005.10, Jeju Korea, pp134-139

CHE W., LIU T. AND LI S. 2005b. Automatic Entity Relation Extraction. *Journal of Chinese Information Processing*. 2005, 19(2):1-6

CHEN J, JI D, TAN C, AND NIU Z. 2006a. Unsupervised Relation Disambiguation Using Spectral Clustering. *in Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics (COLING-ACL'06).* 89-96. Sydney, Australia,

CHEN J, JI D, TAN C, NIU Z. 2006b Relation Extraction Using Label Propagation Based Semi-supervised Learning. *in Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics (COLING-ACL'06),* 129-136. Sydney, Australia

CHEN Y., LI W., LIU Y., ZHENG D., AND ZHAO T. 2010. Exploring Deep Belief Network for Chinese Relation Extraction, *in Proceedings of the Joint Conference on Chinese Language Processing (CLP'10)*, Beijing, China, August 28-29, 2010.

CORTES C. AND VAPNIK V. 1995. Support-vector network. *Machine Learning* 20,273-297.

CULOTTA A. AND SORENSEN J. 2004. Dependency Tree Kernels for Relation Extraction, *In Proceedings of the 42th Annual Meeting of the Association for Computer Linguistics (ACL'04)*, pages 423-429.

CULOTTA A., MCCALLUM A., BETZ J. 2006. Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text. *In Proceedings of the Loint Human Language Technology Conference/ Annula Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'06). 2006*

FORMAN G.. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3: 1289–1305, 2003.

HADDOW B. 2008. Using automated feature optimisation to create an adaptable relation extraction system. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing* (BioNLP '08). Association for Computational Linguistics, Morristown, NJ, USA, 19-27.

HOFMANN T. 1999. Probabilistic Latent Semantic Indexing. *In Proc. of The 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR* 1999): 50-57

HUANG R. H., SUN L. AND FENG Y.Y. 2008. Study of Kernel-Based Methods for Feature space for Relation Extraction, *In Proc. of 4th Asia Infomation Retrieval Symposium (AIRS 2008)*, pp598-604. Harbin, China. Lecture Notes in Computer Science 4993 Springer 2008

JIANG J., ZHAI. C. 2007. A Systematic Exploration of the Feature Space for Relation Extraction. *In proceedings of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007),* pages 113-120.

JOACHIMS T. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. *In Proceedings of European Conference on Machine Learning (ECML'98).*

KAMBHATLA N. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. *In Proceedings of the 42th Annual Meeting of the Association for Computer Linguistics (ACL'04),* pages 178-181.

KAMBHATLA N. 2006. Minority Vote: At-Least-N Voting Improves Recall for Extracting Relations. *in Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics (COLING-ACL'06),* 460-466

KATRENKO S., ADRIAANS P., AND VAN SOMEREN M. 2010. Using local alignments for relation recognition. *J. Artif. Int. Res. 38, 1* (May 2010), 1-48.

LI W., QIAN D., LU Q., YUAN C. 2007. Detecting, categorizing and clustering entity mentions in Chinese text. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*, 647-654, Amsterdam, The Netherlands, July 23-27,

LI W., ZHANG P., WEI F., LU Q. HOU Y. 2008 A Novel Feature-based Approach to Chinese Entity Relation Extraction. *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08),* 89-92. June 15-20, 2008, Columbus, Ohio, USA

MANEVITZ M. L. AND YOUSEF M. 2001. One-class svms for document classification. *Journal of Machine Learning Research, 2(Dec)*:139-154, 2001.

MILLER S., FOX H., RAMSHAW L. AND WEISCHEDEL R. 2000. A novel use of statistical parsing to extract information from text. *In Proceedings of 6th Applied Natural Language Processing Conference*. 29 April- 4 May 2000, Seattle, USA

MIYAO Y., SÆTRE R., SAGAE K., MATSUZAKI T. AND TSUJII J. 2008 . Task-oriented Evaluation of Syntactic Parsers and Their Representations (2008) *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08),* 46-54. June 15-20, 2008, Columbus, Ohio, USA

NAKOV P. AND HEARST M. 2008. Solving Relational Similarity Problems Using the Web as a Corpus. *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08),* 452-460. June 15-20, 2008, Columbus, Ohio, USA

SRA S. (2006). Efficient Large Scale Linear Programming Support Vector Machines. *ECML 2006*: 767-774

TAKAAKI H., SATOSHI S. AND RALPH G. 2004. Discovering Relations among Named Entities from Large Corpora*, In Proceedings of the 42th Annual Meeting of the Association for Computer Linguistics (ACL'04).* Barcelona, Spain.

WANG T. AND LI Y. 2006 Automatic Extraction of Hierarchical Relations from texts. *In Proceedings of the Third European Semantic Web Conference (ESWC 2006),* Budva.

WU F. AND WELD S. D. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (ACL'10). Association for Computational Linguistics, Morristown, NJ, USA, 118-127.

YANG, Y., & PEDERSEN, J. O. 1997. A comparative study on feature selection in text categorization. *In Proc. of the Fourteenth International Conference on Machine Learning (ICML 1997)* (pp. 412-420). Nashville, Tennessee, USA

ZELENKO D. AONE C. AND RICHARDELLA A. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research* 3:1083-1106

ZHANG J., OUYANG Y., LI W., AND HOU Y. 2009. A Novel Composite Kernel Approach to Chinese Entity Relation Extraction. In *Proceedings of 22nd International Conference on Computer Processing of Oriental Languages (ICCPOL 2009),* 236-247. Hong Kong, March 26-27, 2009.

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 the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics (COLING-ACL'06),* pages 825-832. Sydney, Australia

ZHANG Z. 2004. Weakly-supervised relation classification for Information Extraction, *In Proceedings of ACM 13th conference on Information and Knowledge Management (CIKM'2004).* 8-13 Nov 2004. Washington D.C.,USA.

ZHOU G., SU J., ZHANG J., AND ZHANG M. 2005. Exploring Various Knowledge in Relation Extraction. *In Proceedings of the 43rd Annual Meeting of the Association for Computer Linguistics (ACL'05),* pages 427-434.

ZHOU G., ZHANG M., JI D. AND ZHU Q. 2007a. Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*, pages 728-736.

ZHOU G. AND ZHANG M. 2007b. Extracting relation information from text documents by exploring various types of knowledge. *Inf. Process. Manage.* 43(4): 969-982 (2007)

ZHOU J., XU Q., CHEN J., AND QU W. 2009a. A Multi-view Approach for Relation Extraction. In *Proceedings of the International Conference on Web Information Systems and* Mining (WISM '09), Wenyin Liu, Xiangfeng Luo, Fu Lee Wang, and Jingsheng Lei (Eds.). Springer-Verlag, Berlin, Heidelberg

ZHOU G., QIAN L., AND ZHU Q. 2009b. Label propagation via bootstrapped support vectors for semantic relation extraction between named entities. *Comput. Speech Lang.*23, 4

ZHOU G., QIAN L., FAN J. 2010. Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Inf. Sci.* 180(8): 1313-1325 (2010)