A Novel Machine Learning Approach for the Identification of Named Entity Relations

Tianfang Yao
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, 200030, China
yao-tf@cs.sjtu.edu.cn

Hans Uszkoreit
Department of Computational Linguistics and Phonetics
Saarland University
Saarbruecken, 66041, Germany
uszkoreit@coli.uni-sb.de

Abstract

In this paper, a novel machine learning approach for the identification of named entity relations (NERs) called positive and negative case-based learning (PNCBL) is proposed. It pursues the improvement of the identification performance for NERs through simultaneously learning two opposite cases and automatically selecting effective multi-level linguistic features for NERs and non-NERs. This approach has been applied to the identification of domain-specific and cross-sentence NERs for Chinese texts. The experimental results have shown that the overall average recall, precision, and F-measure for 14 NERs are 78.50%, 63.92% and 70.46% respectively. In addition, the above F-measure has been enhanced from 63.61% to 70.46% due to adoption of both positive and negative cases.

1 Introduction

The investigation for Chinese information extraction is one of the topics of the project COLLATE dedicated to building up the German Competence Center for Language Technology. After accomplishing the task concerning named entity (NE) identification, we go on studying identification issues for named entity relations (NERs). As an initial step, we define 14 different NERs based on six identified NEs in a sports domain based Chinese named entity recognition system (Yao et al., 2003). In order to learn NERs, we annotate the output texts from this system with XML. Meanwhile, the NER annotation is performed by an interactive mode.

The goal of the learning is to capture valuable information from NER and non-NER patterns, which is implicated in different features and helps us identify NERs and non-NERs. Generally speaking, because not all features we predefine are important for each NER or non-NER, we should distinguish them by a reasonable measure mode. According to the selection criterion we propose - self-similarity, which is a quantitative measure for the concentrative degree of the same kind of NERs or non-NERs in the corresponding pattern library, the effective feature sets - general-character feature (GCF) sets for NERs and individual-character feature (ICF) sets for non-NERs are built. Moreover, the GCF and ICF feature weights serve as a pro portion determination of the features' degree of importance for identifying NERs against non-NERs. Subsequently, identification thresholds can also be determined.

In the NER identification, we may be confronted with the problem that an NER candidate in a new case matches more than one positive case, or both positive and negative cases. In such situations, we have to employ a vote to decide which existing

Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP, pages 1–8,
Ann Arbor, June 2005.

case environment is more similar to the new case. In addition, a number of special circumstances should be also considered, such as relation conflict and relation omission.

2 Definition of Relations

An NER may be a modifying / modified, dominating / dominated, combination, collocation or even cross-sentence constituent relationship between NEs. Considering the distribution of different kinds of NERs, we define 14 different NERs based on six identified NEs in the sports domain shown in Table 1.

NER Category	Explanation
PS_TM	The membership of a person in a sports team.
PS_CP	A person takes part in a sports competition.
PS_CPC	The origin location of a person.
PS_ID	A person and her / his position in a sports team or other occasions.
HT_VT	The home and visiting teams in a sports competition.
WT_LT	The winning and losing team name in a sports match.
DT_DT	The names of two teams which draw a match.
TM_CP	A team participates in a sports competition.
TM_CPC	It indicates where a sports team comes from.
ID_TM	The position of a person employed by a sports team.
CP_DA	The staged date for a sports competition.
CP_TI	The staged time for a sports competition.
CP_LOC	It gives the location where a sports match is held.
LOC_CPC	The location ownership (LOC belongs to CPC).

Table 1. NER Category

In order to further indicate the positions of NEs in an NER, we define a general frame for the above NERs and give the following example using this description:

Definition 1 (General Frame of NERs):
NamedEntityRelation (NamedEntity$_1$, ParagraphSentenceNamedEntityNo$_1$; NamedEntity$_2$, ParagraphSentenceNamedEntityNo$_2$)

Example 1:

广东宏远队[1]客场以 3 比 0 击败广州太阳神队。
The Guangdong Hongyuan Team defeated the Guangzhou Taiyangshen Team by 3: 0 in the guest field.

In the sentence we observe that there exist two NERs. According to the general frame, the first NER description is HT_VT(广州太阳神队 (Guangzhou Taiyangshen Team), 1-1-2; 广东宏远队(Guangdong Hongyuan Team), 1-1-1) and the other is WT_LT(广东宏远队 (Guangdong Hongyuan Team), 1-1-1; 广州太阳神(Guangzhou Taiyangshen Team), 1-1-2).

In this example, two NERs represent dominating / dominated and collocation relationships separately: namely, the first relation HT_VT gives the collocation relationship for the NE "Guangdong Hongyuan Team" and the noun "guest field". This implies that "Guangdong Hongyuan Team" is a guest team. Adversely, "Guangzhou Taiyangshen Team" is a host team; the second relation WT_LT indicates dominating / dominated relationship between "Guangdong Hongyuan Team" and "Guangzhou Taiyangshen Team" by the verb "defeat". Therefore, "Guangdong Hongyuan Team" and "Guangzhou Taiyangshen Team" are the winning and losing team, respectively.

3 Positive and Negative Case-Based Learning

The positive and negative case-based learning (PNCBL) belongs to supervised statistical learning methods (Nilsson, 1996). Actually, it is a variant of memory-based learning (Stanfill and Waltz, 1986; Daelemans, 1995; Daelemans et al., 2000). Unlike memory-based learning, PNCBL does not simply store cases in memory but transforms case forms into NER and non-NER patterns. Additionally, it stores not only positive cases, but also negative ones. Here, it should be clarified that the negative case we mean is a case in which two or more NEs do not stand in any relationships with each other, i.e, they bear non-relationships which are also investigated objects in which we are interested.

During the learning, depending on the average similarity of features and the self-similarity of NERs (also non-NERs), the system automatically selects general or individual-character features (GCFs or ICFs) to construct a feature set. It also determines different feature weights and identification thresholds for different NERs or non-NERs. Thus, the learning results provide an identification references for the forthcoming NER identification.

3.1 Relation Features

Relation features, by which we can effectively identify different NERs, are defined for capturing critical information of the Chinese language. According to the features, we can define NER / non-

[1] The underlining of Chinese words means that an NE consists of these words.

NER patterns. The following essential factors motivate our definition for relation features:

- The relation features should be selected from multiple linguistic levels, i.e., morphology, grammar and semantics (Cardie, 1996);
- They can help us to identify NERs using positive and negative case-based machine learning as their information do not only deal with NERs but also with non-NERs; and
- They should embody the crucial information of Chinese language processing (Dang et al., 2002), such as word order, the context of words, and particles etc.

There are a total of 13 relation features shown in Table 2, which are empirically defined according to the above motivations. It should be explained that in order to distinguish feature names from element names of the NER / non-NER patterns, we add a capital letter "F" in the ending of feature names. In addition, a sentence group in the following definitions can contain one or multiple sentences. In other words, a sentence group must end with a stop, semicolon, colon, exclamation mark, or question mark.

Feature Category	Explanation
SGTF	The type of a sentence group in which there exists a relation.
NESPF	The named entities of a relevant relation are located in the same sentence or different sentences.
NEOF	The order of the named entities of a relevant relation.
NEVPF	The relative position between the verbs and the named entities of a relevant relation. The verbs of a relevant relation mean that they occur in a sentence where the relation is embedded.
NECF	The context of named entities. The context only embodies a word or a character preceding or following the current named entity.
VSPF	The verbs are located in the same sentence or different sentences in which there is a relevant relation.
NEPPOF	The relative order between parts-of-speech of particles and named entities. The particles occur within the sentences where the relation is embedded.
NEPF	The parts-of-speech of the named entities of a relevant relation.
NECPF	The parts-of-speech of the context for the named entities associated with a relation.
SPF	The sequence of parts-of-speech for all sentence constituents within a relation range.
VVF	The valence expression of verbs in the sentence(s) where there is a relation embedded.
NECTF	The concepts of the named entities of a relevant relation from HowNet (Dong and Dong, 2000).
VCTF	The concepts of the verbs of a relevant relation from HowNet.

Table 2. Feature Category

In 13 features, three features (NECF, NECPF and NEPF) belong to morphological features, three features (NEOF, SPF and SGTF) are grammatical features, four features (NEPPOF, NESPF, NEVPF and VSPF) are associated with not only morphology but also grammar, and three features (NECTF, VCTF and VVF) are semantic features.

Every feature describes one or more properties of a relation. Through the feature similarity calculation, the quantitative similarity for two relations can be obtained, so that we can further determine whether a candidate relation is a real relation. Therefore, the feature definition plays an important role for the relation identification. For instance, NECF can capture the noun 客场 (the guest field, it means that the guest team attends a competition in the host team's residence.) and also determine that the closest NE by this noun is 广东宏远队 (the Guangdong Hongyuan Team). On the other hand, NEOF can fix the sequence of two relation-related NEs. Thus, another NE 广州太阳神队 (the Guangzhou Taiyangshen Team) is determined. Therefore, these two features reflect the properties of the relation HT_VT.

3.2 Relation and Non-Relation Patterns

A relation pattern describes the relationships between an NER and its features. In other words, it depicts the linguistic environment in which NERs exist.

Definition 2 (Relation Pattern): A relation pattern (RP) is defined as a 14-tuple: RP = (NO, RE, SC, SGT, NE, NEC, VERB, PAR, NEP, NECP, SP, VV, NECT, VCT) where *NO* represents the number of a RP; *RE* is a finite set of relation expressions; *SC* is a finite set for the words in the sentence group except for the words related to named entities; *SGT* is a sentence group type; *NE* is a finite set for named entities in the sentence group; *NEC* is a finite set that embodies the context of named entities; *VERB* is a finite set that includes the sequence numbers of verbs and corresponding verbs; *NEP* is a finite set of named entities and their POS tags; *NECP* is a finite set which contains the POS tags of the context for named entities; *SP* is a finite set in which there are the sequence numbers as well as corresponding POS tags and named entity numbers in a sentence group; *VV* is a finite set comprehending the posi-

tion of verbs in a sentence and its valence constraints from Lexical Sports Ontology which is developed by us; *NECT* is a finite set that has the concepts of named entities in a sentence group; and *VCT* is a finite set which gives the concepts of verbs in a sentence group.

Example 2:

据新华社北京 3 月 2 6 日电全国足球甲 B 联赛今天进行了第二轮赛事的 5 场比赛，广东宏远队客场以 3 比 0 击败广州太阳神队，成为唯一一支两战全胜的队伍，暂居积分榜榜首。

According to the news from Xinhua News Agency Beijing on March 26th: National Football Tournament (the First B League) today held five competitions of the second round, The Guangdong Hongyuan Team defeats the Guangzhou Taiyangshen Team by 3: 0 in the guest field, becoming the only team to win both matches, and temporarily occupying the first place of the entire competition.

Relation Pattern:
NO = 34;
RE = {(CP_DA, NE1-3, NE1-2), (CP_TI, NE1-3, NE1-4), …, (WT_LT, NE2-1, NE2-2)}
SC = {(1, 据, according_to, Empty, AccordingTo), (2, 新华社, Xinhua/Xinhua_News_agency, Empty, institution/news/ProperName/China), …, (42, 。, ., Empty, {punc})};
SGT = multi-sentences;
NE = {(NE1-1, 3, LN, {(1, 北京)}), (NE1-2, 4, Date, {(1, 3), (2, 月), (3, 2 6), (4, 日)}), ..., (NE2-2, 26, TN, {(1, 广州), (2, 太阳神), (3, 队)})};
NEC = {(NE1-1, 新华社, 3), (NE1-2, 北京, 电), ..., (NE2-2, 击败, ,) };
VERB = {(8, 进行), (25, 击败), ..., (39, 居)}
PAR = {(1, 据), (9, 了), ..., (38, 暂)};
NEP = {(NE1-1, {(1, N5)}), (NE1-2, {(1, M), (2, N), (3, M), (4, N)}), ..., (NE2-2, {(1, N5), (2, N), (3, N)})};
NECP = {(NE1-1, N, M), (NE1-2, N5, N), …, (NE2-2, V, W)};
SP = {(1, P), (2, N), (3, NE1-1), ..., (42, W)};
VV = {(V_8, {Agent|fact/compete|CT, -Time|time|DT}), (V_25, {Agent|human/mass|TN, Patient|human/mass|TN}),..., (V_39, {Agent|human/sport|PN, Agent|human/mass|TN})};
NECT = {(NE1-1, place/capital/ProperName/China), (NE1-2, Empty+celestial/unit/time+Empty+ celestial/time/time/morning), …, (NE2-2, place/city/ProperName/China+ Empty+community/human/mass)};
VCT = {(V_8, GoForward/GoOn/Vgoingon), (V_25, defeat), …, (V_39, reside/situated)}

Analogous to the definition of the relation pattern, a non-relation pattern is defined as follows:

Definition 3 (Non-Relation Pattern): A non-relation pattern (NRP) is also defined as a 14-tuple: NRP = (NO, NRE, SC, SGT, NE, NEC, VERB, PAR, NEP, NECP, SP, VV, NECT, VCT), where NRE is a finite set of non-relation expressions which specify the nonexistent relations in a sentence group. The definitions of the other elements are the same as the ones in the relation pattern. For example, if we build an NRP for the above sentence group in Example 2, the NRE is listed in the following:

NRE = {(CP_LOC, NE1-3, NE1-1), (TM_CPC, NE2-1, NE1-1), ..., (DT_DT, NE2-1, NE2-2)}

In this sentence group, the named entity (CT) 全国足球甲 B 联赛 (National Football Tournament (the First B League)) does not bear the relation CP_LOC to the named entity (LN) 北京 (Beijing). This LN only indicates the release location of the news from Xinhua News Agency.

As supporting means, the non-NER patterns also play an important role, because in the NER pattern library we collect sentence groups in which the NER exists. If a sentence group only includes non-NERs, obviously, it is excluded from the NER pattern library. Thus the impact of positive cases cannot replace the impact of negative cases. With the help of non-NER patterns, we can remove misidentified non-NERs and enhance the precision of NER identification.

3.3 Similarity Calculation

In the learning, the similarity calculation is a kernel measure for feature selection.

Definition 4 (Self-Similarity): The self-similarity of a kind of NERs or non-NERs in the corresponding library can be used to measure the concentrative degree of this kind of relations or non-relations. The value of the self-similarity is between 0 and 1. If the self-similarity value of a kind of relation or non-relation is close to 1, we can say that the concentrative degree of this kind of relation or non-relation is very "tight". Conversely, the concentrative degree of that is very "loose".

The calculation of the self-similarity for the same kind of NERs is equal to the calculation for the average similarity of the corresponding relation features. Suppose R(i) is a defined NER in the NER set ($1 \le i \le 14$). The average similarity for this kind of NERs is defined as follows:

$$Sim_{average}(R(i)) = \frac{\sum_{1 \le j,\, k \le m;\, j \ne k} Sim\,(R(i)_j, R(i)_k)}{Sum_{relation\_pair}(R(i)_j, R(i)_k)} \quad (1)$$

where Sim $(R(i)_j, R(i)_k)$ denotes the relation similarity between the same kind of relations, $R(i)_j$ and

$R(i)_k$. $1 \le j, k \le m, j \ne k$; m is the total number of the relation R(i) in the NER pattern library. The calculation of $Sim(R(i)_j, R(i)_k)$ depends on different features. $Sum_{relation\_pair}(R(i)_j, R(i)_k)$ is the sum of calculated relation pair number. They can be calculated using the following formulas:

$$Sim(R(i)_j, R(i)_k) = \frac{\sum_{t=1}^{Sumf} Sim(R(i)_j, R(i)_k)(f_t)}{Sum_f} \quad (2)$$

$$Sum_{relation\_pair}(R(i)_j, R(i)_k) = \begin{cases} 1 & m = 2 \\ \frac{m!}{(m-2)! * 2!} & m > 2 \end{cases} \quad (3)$$

In the formula (2), f_t is a feature in the feature set ($1 \le t \le 13$). Sum_f is the total number of features. The calculation formulas of $Sim(R(i)_j, R(i)_k)(f_t)$ depend on different features. For example, if f_t is equal to NECF, $Sim(R(i)_j, R(i)_k)(f_t)$ is shown as follows:

$$Sim(X(i)_j, X(i)_k)(NECF) = \begin{cases} 1 & \text{if all contexts of named entities for two relations are the same} \\ 0.75 & \text{if only a preceding or following context is not the same} \\ 0.5 & \text{if two preceding and / or following contexts are not the same} \\ 0.25 & \text{if three preceding and / or following contexts are not the same} \\ 0 & \text{if all contexts of named entities for two relations are not the same} \end{cases} \quad (4)$$

Notice that the similarity calculation for non-NERs is the same as the above calculations.

Before describing the learning algorithm, we want to define some fundamental conceptions related to the algorithm as follows:

Definition 5 (General-Character Feature): If the average similarity value of a feature in a relation is greater than or equal to the self-similarity of this relation, it is called a General-Character Feature (GCF). This feature reflects a common characteristic of this kind of relation.

Definition 6 (Individual-Character Feature): An Individual-Character Feature (ICF) means its average similarity value in a relation is less than or equal to the self-similarity of this relation. This feature depicts an individual property of this kind of relation.

Definition 7 (Feature Weight): The weight of a selected feature (GCF or ICF) denotes the important degree of the feature in GCF or ICF set. It is used for the similarity calculation of relations or non-relations during relation identification.

$$f(s)_w(R(i)) = \frac{Sim_{average}f(s)(R(i))}{\sum_{t=1}^{n} Sim_{average}f(t)(R(i))} \quad (5)$$

where R(i) is a defined relation in the NER set ($1 \le i \le 14$); n is the size of selected features, $1 \le s, t \le n$; and

$$Sim_{average}f(s)(R(i)) = \frac{\sum_{1 \le j, k \le m; j \ne k} Sim(R(i)_j, R(i)_k)(f(s))}{Sum_{relation\_pair}(R(i)_j, R(i)_k)} \quad (6)$$

$Sim(R(i)_j, R(i)_k)(f(s))$ computes the feature similarity of the feature f(s) between same kinds of relations, $R(i)_j$ and $R(i)_k$. $1 \le j, k \le m, j \ne k$; m is the total number of the relation R(i) in the NER pattern library. $Sum_{relation\_pair}(R(i)_j, R(i)_k)$ is the sum of calculated relation pair numbers, which can be calculated by the formula (3).

Definition 8 (Identification Threshold): If a candidate relation is regarded as a relation in the relation pattern library, the identification threshold of this relation indicates the minimal similarity value between them. It is calculated by the average of the sum of average similarity values for selected features:

$$IdenThrh(R(i)) = \frac{\sum_{t=1}^{n} Sim_{average}f(t)(R(i))}{n} \quad (7)$$

where n is the size of selected features, $1 \le t \le n$.

Finally, the PNCBL algorithm is described as follows:

1) Input annotated texts;
2) Transform XML format of texts into internal data format;
3) Build NER and non-NER patterns;
4) Store both types of patterns in hash tables and construct indexes for them;

5) Compute the average similarity for features and self-similarity for NERs and non-NERs;
6) Select GCFs and ICFs for NERs and non-NERs respectively;
7) Calculate weights for selected features;
8) Decide identification thresholds for every NER and non-NER;
9) Store the above learning results.

4 Relation Identification

Our approach to NER identification is based on PNCBL, it can utilize the outcome of learning for further identifying NERs and removing non-NERs.

4.1 Optimal Identification Tradeoff

During the NER identification, the GCFs of NER candidates match those of all of the same kind of NERs in the NER pattern library. Likewise, the ICFs of NER candidates compare to those of non-NERs in the non-NER pattern library. The computing formulas in this procedure are listed as follows:

$$\mathrm{Sim}\,(R(i)_{can}, R(i)_{j1}) = \sum_{k1=1}^{\mathrm{Sum}(GCF)_i} \{\, w_i\,(GCF_{k1}) * \mathrm{Sim}\,(R(i)_{can}, R(i)_{j1})\,(GCF_{k1}) \,\} \quad (8)$$

and

$$\mathrm{Sim}\,(R(i)_{can}, NR(i)_{j2}) = \sum_{k2=1}^{\mathrm{Sum}(ICF)_i} \{\, w_i\,(ICF_{k2}) * \mathrm{Sim}\,(R(i)_{can}, NR(i)_{j2})\,(ICF_{k2}) \,\} \quad (9)$$

where R(i) represents the NER_i, and NR(i) expresses the non-NER_i, $1 \le i \le 14$. $R(i)_{can}$ is defined as a NER_i candidate. $R(i)_{j1}$ and $NR(i)_{j2}$ are the j1-th NER_i in the NER pattern library and the j2-th non-NER_i in the non-NER pattern library. $1 \le j1 \le$ Sum (R(i)) and $1 \le j2 \le$ Sum (NR(i)). Sum (R(i)) and Sum (NR(i)) are the total number of R(i) in the NER pattern library and that of NR(i) in non-NER pattern library respectively. w_i (GCF_{k1}) and w_i (ICF_{k2}) mean the weight of the k1-th GCF for the NER_i and that of the k2-th ICF for the non-NER_i. Sum $(GCF)_i$ and Sum $(ICF)_i$ are the total number of GCF for NER_i and that of ICF for non-NER_i separately.

In matching results, we find that sometimes the similarity values of a number of NERs or non-NERs matched with NER candidates are all more than the identification threshold. Thus, we have to utilize a voting method to achieve an identification tradeoff in our approach. For an optimal tradeoff, we consider the final identification performance in two aspects: i.e., recall and precision. In order to enhance recall, as many correct NERs should be captured as possible; on the other hand, in order to increase precision, misidentified non-NERs should be removed as accurately as possible.

The voting refers to the similarity calculation results between an NER candidate and NER / non-NER patterns. It pays special attention to circumstances in which both results are very close. If this happens, it exploits multiple calculation results to measure and arrive at a final decision. Additionally, notice that the impact of non-NER patterns is to restrict possible misidentified non-NERs. On the other hand, the voting assigns different thresholds to different NER candidates (e.g. HT_VT, WT_LT, and DT_DT or other NERs). Because the former three NERs have the same kind of NEs, the identification for these NERs is more difficult than for others. Thus, when voting, the corresponding threshold should be set more strictly.

4.2 Resolving NER Conflicts

In fact, although the voting is able to use similarity computing results for yielding an optimal tradeoff, there still remain some problems to be resolved. The relation conflict is one of the problems, which means that contradictory NERs occur in identification results. For example:

(i) The same kind of relations with different argument position: e.g., the relations HT_VT,

> HT_VT(ne1, no1; ne2, no2) and HT_VT(ne2, no2; ne1, no1) occur in an identification result at the same time.

(ii) The different kinds of relations with same or different argument positions: e.g., the relations WT_LT and DT_DT,

> WT_LT(ne1, no1; ne2, no2) and DT_DT(ne1, no1; ne2, no2) appear simultaneously in an identification result.

The reason for a relation conflict lies in the simultaneous and successful matching of a pair of NER candidates whose NEs are the same kind. They do not compare and distinguish themselves further. Considering the impact of NER and non-NER patterns, we organize the conditions to remove one of the relations, which has lower average similarity value with NER patterns or higher average similarity value with non-NER patterns.

4.3 Inferring Missing NERs

Due to a variety of reasons, some relations that should appear in an identification result may be missing. However, we can utilize some of the identified NERs to infer them. Of course, the prerequisite of the inference is that we suppose identified NERs are correct and non-contradictory. For all identified NERs, we should first examine whether they contain missing NERs. After determining the type of missing NERs, we may infer them - containing the relation name and its arguments. For instance, in an identification result, two NERs are:

PS_ID (ne1, no1; ne2, no2) and PS_TM (ne1, no1; ne3, no3)

In the above NER expressions, ne1 is a personal name, ne2 is a personal identity, and ne3 is a team name, because if a person occupies a position, i.e., he / she has a corresponding identity in a sports team, that means the position or identity belongs to this sports team. Accordingly, we can infer the following NER:

ID_TM (ne2, no2; ne3, no3)

5 Experimental Results and Evaluation

The main resources used for learning and identification are NER and non-NER patterns. Before learning, the texts from the Jie Fang Daily[2] in 2001 were annotated based on the NE identification. During learning, both pattern libraries are established in terms of the annotated texts and Lexical Sports Ontology. They have 142 (534 NERs) and 98 (572 non-NERs) sentence groups, respectively.

To test the performance of our approach, we randomly choose 32 sentence groups from the Jie Fang Daily in 2002, which embody 117 different NER candidates.

For evaluating the effects of negative cases, we made two experiments. Table 3 shows the average and total average recall, precision, and F-measure for the identification of 14 NERs only by positive case-based learning. Table 4 demonstrates those by PNCBL. Comparing the experimental results, among 14 NERs, the F-measure values of the seven NERs (PS_ID, ID_TM, CP_TI, WT_LT, PS_CP, CP_DA, and DT_DT) in Table 4 are higher than those of corresponding NERs in Table 3; the F-measure values of three NERs (LOC_CPC, TM_CP, and PS_CP) have no variation; but the F-measure values of other four NERs (PS_TM, CP_LOC, TM_CPC, and HT_VT) in Table 4 are lower than those of corresponding NERs in Table 3. This shows the performances for half of NERs are improved due to the adoption of both positive and negative cases. Moreover, the total average F-measure is enhanced from 63.61% to 70.46% as a whole.

Relation Type	Average Recall	Average Precision	Average F-measure
LOC_CPC	100	91.67	95.65
TM_CP	100	87.50	93.33
PS_ID	100	84.62	91.67
PS_TM	100	72.73	84.21
CP_LOC	88.89	69.70	78.13
ID_TM	90.91	66.67	76.93
CP_TI	83.33	71.43	76.92
PS_CP	60	75	66.67
TM_CPC	100	42.50	59.65
HT_VT	71.43	38.46	50
WT_LT	80	30.77	44.45
PS_CPC	33.33	66.67	44.44
CP_DA	0	0	0
DT_DT	0	0	0
Total Ave.	**71.99**	**56.98**	**63.61**

Table 3: Identification Performance for 14 NERs only by Positive Case-Based Learning

Relation Type	Average Recall	Average Precision	Average F-measure
LOC_CPC	100	91.67	95.65
TM_CP	100	87.50	93.33
CP_TI	100	75	85.71
PS_CPC	100	68.75	81.48
ID_TM	90.91	68.19	77.93
PS_ID	72.22	81.67	76.65
CP_LOC	88.89	66.67	76.19
PS_TM	80	65	71.72
CP_DA	100	50	66.67
DT_DT	66.67	66.67	66.67
PS_CP	60	75	66.67
WT_LT	60	37.50	46.15
HT_VT	42.86	30	35.30
TM_CPC	37.50	31.25	34.09
Total Ave.	**78.50**	**63.92**	**70.46**

Table 4: Identification Performance for 14 NERs by PNCBL

Finally, we have to acknowledge that it is difficult to compare the performance of our method to others because the experimental conditions and corpus domains of other NER identification efforts are quite different from ours. Nevertheless, we would like to use the performance of Chinese NER identification using memory-based learning (MBL) (Zhang and Zhou, 2000) for a comparison with our approach in Table 5. In the table, we select similar NERs in our domain to correspond to the three types of the relations (*employee-of*, *product-of*, and *location-of*). From the table we can deduce that the

[2] This is a local newspaper in Shanghai, China.

identification performance of relations for PNCBL is roughly comparable to that of the MBL.

Method	Relation Type	Recall	Precision	F-measure
MBL&I	employee-of	75.60	92.30	83.12
	product-of	56.20	87.10	68.32
	location-of	67.20	75.60	71.15
PNCBL&I	PS_TM	80	65	71.72
	PS_CP	60	75	66.67
	PS_ID	72.22	81.67	76.65
	ID_TM	90.91	68.19	77.93
	TM_CP	100	87.50	93.33
	CP_LOC	88.89	66.67	76.19
	PS_CPC	100	68.75	81.48
	TM_CPC	37.50	31.25	34.09

Table 5: Performances for Relation Identification (PNCBL&I vs. MBL&I)

6 Conclusion

In this paper, we propose a novel machine learning and identification approach PNCBL&I. This approach exhibits the following advantages: (i) The defined negative cases are used to improve the NER identification performance as compared to only using positive cases; (ii) All of the tasks, building of NER and non-NER patterns, feature selection, feature weighting and identification threshold determination, are automatically completed. It is able to adapt the variation of NER and non-NER pattern library; (iii) The information provided by the relation features deals with multiple linguistic levels, depicts both NER and non-NER patterns, as well as satisfies the requirement of Chinese language processing; (iv) Self-similarity is a reasonable measure for the concentrative degree of the same kind of NERs or non-NERs, which can be used to select general-character and individual-character features for NERs and non-NERs respectively; (v) The strategies used for achieving an optimal NER identification tradeoff, resolving NER conflicts, and inferring missing NERs can further improve the performance for NER identification; (vi) It can be applied to sentence groups containing multiple sentences. Thus identified NERs are allowed to cross sentences boundaries.

The experimental results have shown that the method is appropriate and effective for improving the identification performance of NERs in Chinese.

Acknowledgement

This work is a part of the COLLATE project under contract no. 01INA01B, which is supported by the German Ministry for Education and Research.

References

C. Cardie. 1996. *Automating Feature Set Selection for Case-Based Learning of Linguistic Knowledge*. In Proc. of the Conference on Empirical Methods in Natural Language Processing. University of Pennsylvania, Philadelphia, USA.

W. Daelemans. 1995. *Memory-based lexical acquisition and processing*. In P. Steffens, editor, Machine Translations and the Lexicon, Lecture Notes in Artificial Intelligence, pages 85-98. Springer Verlag. Berlin, Germany.

W. Daelemans, A. Bosch, J. Zavrel, K. Van der Sloot, and A. Vanden Bosch. 2000. *TiMBL: Tilburg Memory Based Learner, Version 3.0, Reference Guide*. Technical Report ILK-00-01, ILK, Tilburg University. Tilburg, The Netherlands. http://ilk.kub.nl/~ilk/papers/ilk0001.ps.gz.

H. Dang, C. Chia, M. Palmer and F. Chiou. 2002. *Simple Features for Chinese Word Sence Disambiguation*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 204-210. Taipei, Taiwan.

Z. Dong and Q. Dong. 2000. *HowNet*. http://www.keenage.com/zhiwang/e_zhiwang.html.

N. Nilsson. 1996. *Introduction to Machine Learning: An Early Draft of a Proposed Textbook*. Pages 175-188. http://robotics.stanford.edu/people/nilsson/mlbook.html.

C. Stanfill and D. Waltz. 1986. *Toward memory-based reasoning*. Communications of the ACM, Vol.29, No.12, pages 1213-1228.

T. Yao, W. Ding and G. Erbach. 2003. *CHINERS: A Chinese Named Entity Recognition System for the Sports Domain*. In: Proc. of the Second SIGHAN Workshop on Chinese Language Processing (ACL 2003 Workshop), pages 55-62. Sapporo, Japan.

Y. Zhang and J. Zhou. 2000. *A trainable method for extracting Chinese entity names and their relations*. In Proc. of the Second Chinese Language Processing Workshop (ACL 2000 Workshop), pages 66-72. Hongkong, China.