A Study of Chinese Named Entity and Relation Identification in a Specific Domain

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften (Dr. -Ing)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Eingereicht von

Tianfang Yao

Saarbrücken, den 31.05.2005

Verfasser:	Tianfang Yao Raum 47 Huai-Hai Straße 453 (mittler) Shanghai 200020 China yao-tf@cs.sjtu.edu.cn
Tag des Kolloquiums:	13.10.2005
Dekan:	Prof. Dr. Jörg Eschmeier
Prüfungsausschuss:	Prof. Dr. Jörg Siekmann (Vorsitzender) Prof. Dr. Hans Uszkoreit (Berichterstatter) Prof. Dr. Wolfgang Wahlster (Berichterstatter) Dr. Günter Neumann

Abstract

This thesis aims at investigating automatic identification of Chinese named entities (NEs) and their relations (NERs) in a specific domain. We have proposed a three-stage pipeline computational model for the error correction of word segmentation and POS tagging, NE recognition and NER identification.

In this model, an error repair module utilizing machine learning techniques is developed in the first stage. At the second stage, a new algorithm that can automatically construct Finite State Cascades (FSC) from given sets of rules is designed. As a supplement, the recognition strategy without NE trigger words can identify the special linguistic phenomena. In the third stage, a novel approach - positive and negative case-based learning and identification (PNCBL&I) is implemented. 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. Further, two other strategies, resolving relation conflicts and inferring missing relations, are also integrated in the identification procedure.

Kurzzusammenfassung

Diese Dissertation ist der Forschung zur automatischen Erkennung von chinesischen Begriffen (named entities, NE) und ihrer Relationen (NER) in einer spezifischen Domäne gewidmet. Wir haben ein Pipelinemodell mit drei aufeinanderfolgenden Verarbeitungsschritten für die Korrektur der Fehler der Wortsegmentation und Wortartmarkierung, NE-Erkennung, und NER-Identifizierung vorgeschlagen.

In diesem Modell wird eine Komponente zur Fehlerreparatur im ersten Verarbeitungsschritt verwirklicht, die ein machinelles Lernverfahren einsetzt. Im zweiten Stadium wird ein neuer Algorithmus, der die Kaskaden endlicher Transduktoren aus den Mengen der Regeln automatisch konstruieren kann, entworfen. Zusätzlich kann eine Strategie für die Erkennung von NE, die nicht durch das Vorkommen bestimmer lexikalischer Trigger markiert sind, die spezielle linguistische Phänomene identifizieren. Im dritten Verarbeitungsschritt wird ein neues Verfahren, das auf dem Lernen und der Identifizierung positiver und negativer Fälle beruht, implementiert. Es verfolgt die Verbesserung der NER-Erkennungsleistung durch das gleichzeitige Lernen zweier gegenüberliegenden Fälle und die automatische Auswahl der wirkungsvollen linguistischen Merkmale auf mehreren Ebenen für die NER und Nicht-NER. Weiter werden zwei andere Strategien, die Lösung von Konflikten in der Relationenerkennung und die Inferenz von fehlenden Relationen, auch in den Erkennungsprozeß integriert.

Ausführliche Zusammenfassung

Wegen der exponentialen Zunahme der unstrukturierten Online-Texte, wird es immer schwieriger, nützliche und relevante Informationen innerhalb einer annehmbaren Zeitbegrenzung zu finden. Die Aufgabe der Informationsextraktion (IE) ist es, relevante Information aus einer grossen Menge freier Texte zu finden um Datenbankeinträge auf eine leistungsfähige und robuste Weise zu finden und zu extrahieren. Obgleich die IE Technologie, die für das Englische und andere westliche Sprachen entwickelt wurde, bemerkenswerte Erfolge erzielt hat, befindet sich die IE-Forschung für das Chinesische noch in einem unreifen Stadium. Insbesondere ist bis lang nur wenig Arbeit in die Forschung zur Erkennung von chinesischen Begriffen und Relationen zwischen den Denotaten solcher Begriffe investiert worden.

Diese Dissertation ist der Forschung zur automatischen Erkennung von chinesischen Begriffen (named entities, NE) und ihrer Relationen (NER) in einer spezifischen Domäne gewidmet. Die betrachtete Domäne sind Fußballberichte. Die sehr freie und mitunter umgangssprachliche Natur der Texte in dieser Domäne erhöht Schwierigkeiten, die sich für deren Verarbeitung ergeben. Es ist bekannt, dass insbesondere für das Chinesische die Qualität der Wortsegmentation und Wortartmarkierung für IE und viele andere text-verstehende Anwendungen ganz entscheidend ist. Das frei verfügbare Werkzeug für die Wortsegmentation und Wortartmarkierung, das von uns benutzt wird, erzeugt zahlreiche Fehler. Um das Resultat der Analyse zu verbessern, haben wir eine transformations-basierte Lernmethode für die Reparatur der Fehler entwickelt, die von der ursprünglichen Wortsegmentation und Wortartmarkierung gemacht werden. Nach der Revision der Resultate der Wortsegmentation und Wortartmarkierung haben wir uns auf die zwei Hauptaufgaben nämlich die flexible und genaue Erkennung von NE und NER konzentriert.

Wir schlagen ein Pipelinemodell mit drei aufeinanderfolgenden Verarbeitungsschritten vor, um diese Probleme zu lösen. In diesem Modell enthält der erste Verarbeitungsschritt die Fehlerkorrektur der Wortsegmentation und die Wortartmarkierung der Resultate. Wie oben erwähnt, haben wir eine Komponente zur Fehlerreparatur verwirklicht, die ein transformationsbasiertes machinelles Lernverfahren einsetzt. Im zweiten Stadium wird ein regelbariertes Verfahren zur automatischen NE-Erkennung angewendet. Die Sätze der Erkennungsregeln für jede NE-Kategorie werden auf der Basis der beobachteten linguistischen Phänomene in Handarbeit formuliert. Unter Zuhilfenahme der Graphentheorie entwerfen wir einen neuen Algorithmus, der die Kaskaden endlicher Transduktoren aus den Mengen der Regeln automatisch konstruieren kann. In diesem Algorithmus enthalten die regulären Ausdrücke nichtatomare Symbole bestehend aus komplexen Contraints anstatt der sonst üblichen atomaren Symbole. So erweitert unser Formalismus den traditionellen Formalismus für die endlichen Transduktoren und macht ihn damit geeigneter für reale Anwendungen in der Sprachverarbeitung. Zusätzlich unterstützt die automatische Berechnung der Kaskaden die Grammatikentwickler beim Entwurf und der Weiterentwicklung der Regeln. Zusätzlich kann eine Strategie für die Erkennung von NE, die nicht durch das Vorkommen bestimmer lexikalischer Trigger markiert sind, die spezielle linguistische Phänomene identifizieren, die diese NE kennzeichnen. Der dritte Verarbeitungsschritt bewirkt die Erkennung der Relationen zwischen NE. Wir schlagen ein neues Verfahren für die NER vor, das auf dem Lernen positiver und negativer Fälle beruht. Während des Lernens der positiven

und negativen Fälle kann dieses Verfahren die Selbstähnlichkeit jeder NER und Nicht-NER in den Musterbibliotheken berechnen und dann wirkungsvolle linguistische Merkmale auf mehreren Ebenen auswählen, welche die relevanten Kriterien für die Erkennung bestimmen. Danach werden die Merkmalsgewichte festgestellt und die Schwellenwerte für ihre Anwendung festgesetzt. Während der Erkennung wird ein neuer Fall mit den erlernten Resultaten abgeglichen und ein optimaler Kmpromiß gefunden, wenn es mehrere Lösungen gibt. Um die NER-Erkennungsleistung zu erhöhen, werden zudem zwei weitere Strategien, die Lösung von Konflikten in der Relationenerkennung und die Inferenz von fehlenden Relationen, auch in den Erkennungsprozeß integriert.

Die oben genannten Methoden und Algorithmen sind im Prototypsystem CHINERIS in Java implementiert. Das System kann 6 Typen von NE und 14 NER in der Domäne des Fußballberichte automatisch erkennen. Es ist sehr leistungsfähig und verfügt zudem über eine ergonomisch gestaltete Benutzerschnittstelle.

Die experimentellen Resultate der Evaluation der ersten Verarbeitungskomponente zeigen, dass die Fehlerreparatur die Genauigkeit der Erkennung von Wortsegmenten und Wortarten bemerkenswert erhöht. Der Nutzen für die beiden folgenden Verarbeitungsschritte ist somit ganz offensichtlich.

Auch die positiven Evaluationsresultate für die anderen beiden Komponenten bezeugen die Effektivität der entwickelten Verfahren. Zudem können der Ansatz und das Verarbeitungsmodell durch den Einsatz der Methoden aus dem maschinellen Lernen und durch die automatischen Berechnung der FST-Kaskaden leicht auf neue Domänen und Aufgaben angepasst werden.

Acknowledgements

The first person I would like to thank is my supervisor, Prof. Hans Uszkoreit. He has supported me in exploring the field of Information Extraction for Chinese throughout my thesis work. His constructive suggestions and warm encouragement have helped me so much in accomplishing this task.

I have been very fortunate to be given the opportunity to work with other colleagues in the COLLATE project group. I would like to thank Gregor Erbach, who encouraged me in my work and gave valuable opinions for each paper I submitted. I am especially grateful to Wei Ding for assisting me with part of the programming work and some experiments. Many thanks to Zhiping Zheng for lots of beneficial discussions. I would also like to thank Prof. Manfred Pinkal, Dorothee Ziegler-Eisele, Silvia Hansen-Schirra and other project members. They have been very supportive and it was a great experience to work with them.

During my thesis work, I was a guest scientist in DFKI-LT for two years. I want to express my heartfelt gratitude to DFKI for its financial support. In addition, I benefited greatly from their laboratory facilities. I would like to say thank you to Feiyu Xu, both for her helpful discussions and her help in coordinating the cooperation between DFKI-LT and SJTU-CS&E. Very special thanks to Markus Becker and Jakub Piskorski for useful thoughts and warmhearted help. Many thanks also to Günter Neumann, Paul Buitelaar, and Thierry Declerck for their valuable suggestions.

Feiyu Xu, Bobbye Pernice, Prof. Hans Uszkoreit and Prof. Wolfgang Wahlster have read the draft of this thesis; I appreciate their help during the thesis revision. Ling Yin, Wei Ding and Qinghui Zeng also read part of the draft.

I am also grateful to Prof. Huanye Sheng, Vice President of Shanghai Jiao Tong University (SJTU), who recommended me as a PhD candidate to the University of Saarland and supported me for my PhD thesis research. Many thanks also to Fang Li and Dongmo Zhang, my colleagues at SJTU, for their help during my thesis work.

I am greatly indebted to many German and Chinese friends who have given me much help in my life. They are Lansun Chen, Ottilie Deutsch, Wei Ding and Feilong Xu, Edith Hartmann, Maja Hein and Franz Hein, Rita Honecker and Rainer Honecker, Huiyan Huang and Qinghui Zeng, Lili Jin and Xiaoqing Shen, Shenfang Kong, Olga Kulicka, Daojun Ni and Xiaomao Li, Hans M. Reckert, Bin Shen, Yuehua Tang and Yeping Guo, Martin Tschaki, Hanliang Wang, Lili Xing and Bin Zhu, Feiyu Xu and Franz-Josef Selgrad, Zhiyong Xu, Ling Yin, Zhiping Zheng, and Yaxi Zhou

My wife Lei Chen and my son Qingchen Yao have continuously encouraged me to pursue my PhD thesis research. My mother and my sister have also unconditionally supported my work. Without their help, I would not have finished this thesis. I am sincerely grateful to them for their indispensable support and help.

Contents

Chapter 1

Introduction

1.1 Motivations and Overview

Information Extraction (*IE*) is an innovative language technology for accurately acquiring crucial information from documents. When beginning my PhD thesis work, I was confronted with the determination of vital three elements with regard to IE investigation, i.e., IE tasks, applied language, as well as application domain.

Generally speaking, the identification of *Named Entity* (*NE*) is a fundamental IE task, which captures some critical combined constituents in sentences, for instance, personal name, date, time, location name, organization name, etc. Based on this processing, the identification of *Named Entity Relation* (*NER*) can give semantic relationships between identified NEs. These relationships depict modifying / modified, dominating / dominated, combination, collocation and even cross-sentence named entity relationships, e.g., person-birthday, person-nationality, organization-location and so on. They can be provided as a resource for other application systems such as question-answering system. Therefore, these two IE tasks are selected as our investigation emphases. In addition, lexical ontology is a lexical concept knowledge base that can be used to provide domain knowledge for these two tasks, for example, the recognition of NEs without trigger words, the determination of NE boundaries, and the provision of conceptual values as well as the computation of the semantic distance for two concepts during identifying NERs. Accordingly, the investigation for ontology used for IE including how to define taxonomy and organize concept category structure is also critical work.

In the past twenty years there have been significant achievements in IE with respect to western languages such as English. However, it is marked by a *lack* of investigation for Chinese IE in some aspects. For example, for the IE tasks proposed by *Message Understanding Conferences* (*MUCs*) (Grishman and Sundheim, 1996; Chinchor, 1998), compared with western languages, Chinese IE primary concentrated on the NE task. Little work regarding Template Element (TE), Template Relation (TR), Coreference (CO), and Scenario Template (ST) has been done. Furthermore, in most of IE models, only a single approach has been adopted. That is to say, the research for *hybrid* methods has possible been neglected. As to IE system, it has been reported that only few IE prototype and application systems can perform *multiple* IE tasks. In general, a processed language has a major influence upon IE methods and technologies, e.g., one of the primary differences between Chinese and western languages is that it gives no evidence of the orthography where the word boundaries are. Thus, a Chinese IE system must necessarily be complicated by a word segmentation component. Therefore, to investigate some special points with regard to the Chinese language in IE and to improve the current research status of Chinese IE, I have chosen the Chinese language as the applied language for my study.

Our research work refers to domain-specific IE. Hence, we picked *sports*, particularly, football competition review, as our application domain. The reasons for this choice are that, first, the types of NEs and NERs are *varied* in the texts; second, there are many *dynamic* descriptions in the texts, which are related to the verbs that are very important for identifying NERs; third, the text style is *diverse*, e.g., the same event is described by different reviews. Thus, we can use texts with different styles for our experiments. These features of domain texts are of great benefit to the IE investigation.

In this thesis, a computational model with respect to Chinese IE will be presented. Three important stages in this model, namely, word processing, named entity identification and named entity relation identification, are highlighted, which also are the kernels of our investigation for Chinese IE. During the research, we have especially concentrated on some interesting critical points such as, improvement for the performance of Chinese word segmentation and part-of-speech tagging, automatic construction of Finite-State Cascades for the identification of named entities, identification for some special constructions of named entities, relationships between named entity relations and linguistic features, joint effects of positive and negative cases and so on. On the theoretical level, these points are related to Chinese computational linguistics, that is, Chinese morphological, grammatical and semantic theories. On the other hand, in language technologies, this model employs annotation, machine learning, shallow parsing, ontology techniques, etc. Consequently, the model is constructed by a hybrid issue including knowledge engineering and automatic training (Appelt and Israel, 1999). A prototype system using this model, *CHINERIS* (*Chinese Named Entity and Relation Identification System*), has been implemented under the project *COLLATE* (Computational Linguistics and Language Technology for Real World Applications) at the Department of Computational Linguistics of Saarland University.

The rest of this chapter is organized as follows. First of all, a computational model of Chinese IE, which adopts a hybrid issue for enhancing the performance of word segmentation and part-of-speech tagging, named entity identification as well as named entity relation identification, is described in Section 1.2. Then we summarize the main achievements and contributions in this thesis. Finally, the organization of the thesis will be specified in Section 1.4.

1.2 A Chinese IE Computational Model

In the previous section, we have discussed the *insufficiencies* in Chinese IE investigation, which are related not only to Chinese linguistic theories but also to language technologies. In order to intensively study Chinese IE, primarily word processing, NE identification, and NER identification, we propose a Chinese IE computational model for fulfilling the above tasks. Our motivations concerning the establishment of the model are:

- Combining the *different* effective techniques in IE model, such as knowledge-base, statistical, machine learning etc;
- Designing a novel IE computational model that can be suitable to different Chinese IE tasks, such as NE and NER identification;
- Making IE systems more *efficient* and *effective*, especially in reliability, portability, and performance.

Concretely speaking, our goals for the construction of this model are:

- Establishing an IE computational model for Chinese texts on the Internet using *hybrid* technologies, which should to a great extent meet the requirements of IE for Chinese texts;
- Implementing a prototype system based on this IE computational model, which extracts Chinese information as *accurately* and *quickly* as possible;
- Evaluating the *performance* of the prototype system in a specific domain.

In the model, the IE processing is divided into three phases: (i) word processing; (ii) NE identification; (iii) NER identification. Figure 1.1 demonstrates a Chinese IE computational model with three stages.

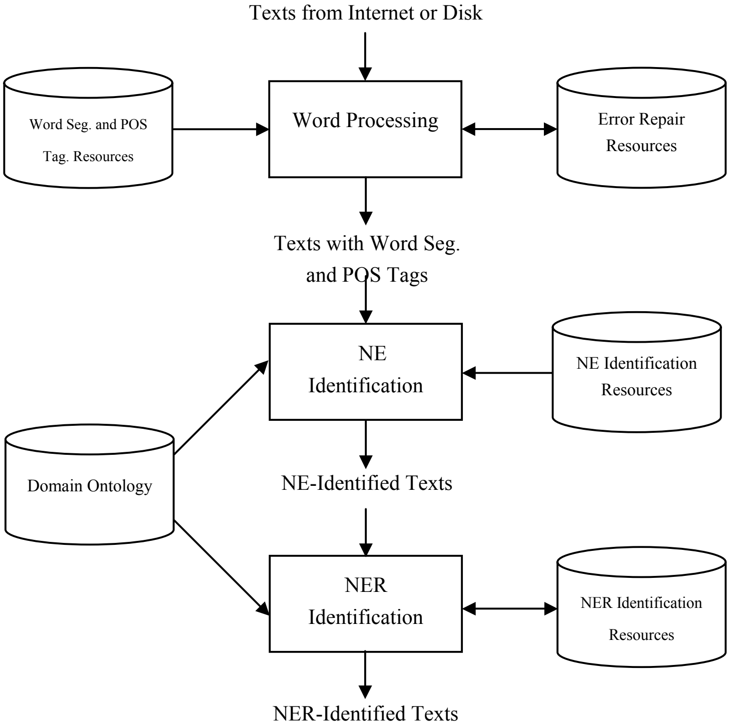

Texts from Internet or Disk
Word Seg. and POS Tag. Resources
Word Processing
Error Repair Resources
Texts with Word Seg. and POS Tags
NE Identification
NE Identification Resources
Domain Ontology
NE-Identified Texts
NER Identification
NER Identification Resources
NER-Identified Texts

Figure 1.1 A Three-Stage Chinese IE Computational Model

This model has a pipeline architecture. In the first stage, that is, word processing including word segmentation and part-of-speech tagging, the *quality* of this stage is critical. For an automatic system, its processing quality (the performance of output results) has considerable influence on the performance of the consequent two stages. In order to generate effective rules to repair different word segmentation and part-of-speech tagging errors, we plan to use an automatic trainable approach to do so. For the second stage, two kinds of NEs will be processed. One is the NEs which have trigger words; the other those without trigger words. For the former NEs, we prepare to adopt knowledge engineering technology, such as *finite-state cascades* (*FSC*) as a shallow parsing mechanism, to *reliably* identify NEs in turn. For the latter NEs, however, we intend to design some special strategies to identify them. NER identification is processed in the last stage, it is based on the output results of the second stage. Because of the diversity and complexity of NERs, for the sake of the identification *portability*, we consider employing an automatic trainable approach.

In the model, in addition to the above technologies, we will adopt an *annotation* technique for machine learning in the last stage and establish a *domain ontology* for semantic and concept processing in the later two stages. Moreover, the integration of the existing system, e.g., word segmentation and part-of-speech tagging system, is also an adopted strategy in our model.

In order to show domain text style and give some examples of NEs and NERs, we list a review from the *Jie Fang Daily*[1] on March 26, 2000, in the following example:

Example 1.1 (Chinese Football Competition Review):

全国女足超级联赛在沪揭幕 上视队一球小胜大连

本报讯上视又利滋女足一球小胜大连凯飞队，为２０００年全国女足超级联赛揭幕战平添了紧张气氛。

昨天下午，万余球迷在上海体育场观看了本年度女足超级联赛开幕式、沪连之战。上届亚军迎战超级联赛新军，两队水平相距不止一个档次。但在前７０分钟时间里，任凭上视姑娘狂轰乱炸，大连城池巍然不动。国家队年轻门将韩文霞“一夫当关，万夫莫开”，多次瓦解了上视队极有威胁的攻门。

直至第７０分钟，上视队才打破僵局。当时，上视队又发起了一次地面进攻，前锋莫晨月在禁区内得球后鱼跃头攻，终于攻破了韩文霞的十指关。

对于上视队在极大部分时间里开花不结果的状况，主教练马良行是这样解释的：“国家队队员归队才三四天，尚未磨合好；我们前锋队员的高度不够，‘上三路’进攻经常受阻；这是今年超级联赛的第一场，有些队员还是紧张、有压力”。当然，他对球队的整体表现及比赛结果感到满意，特别提到了年轻球员朱惠华。

大连队主帅阎群则诙谐地说：“上视队脚下留情，我们只输一个球，当然感到满足啦”

记者杨仁杰

Translation:

National Womens Football Super League Matches Opened in Shanghai: Shanghai Television Team Feebly Beat Dalian by Exactly one Ball

Newspaper News: Shanghai Television You Li Zi Womens Football Team feebly beat the Dalian Kai Fei Team by exactly one ball, which increased the nervous air for the unveiled year 2000 National Womens Football Super League Matches (2000 NWFSLM) in the original calm.

[1] This is a famous local newspaper of Shanghai, China. (http://www.jfdaily.com.cn/)

Yesterday afternoon, over ten thousand of football fans watched the 2000 WFSLM opening ceremony and the fight between Shanghai and Dalian in Shanghai Stadium. The runner-up from last year took on the new army of the Super League Matches. The level of both teams is more than one grade apart. But within 70 minutes, no matter how savagely the girls of Shanghai Television Team bombed, Dalian's city wall and moat stood firm. The young goalkeeper Hang Wenxia of the national team was like "one man can hold out against ten thousand", she disintegrated the attack with extreme threat from the Shanghai Television Team many times. Until the 70th minute, the Shanghai Television Team just broke the ice.

At that time, the Shanghai Television Team launched a ground attack once more, after the forward Mo Chenyue got the ball in the penalty area, she dove and attacked with her head. Finally, she broke through the ten fingers' pass of Hang Wenxia.

With regard to the Shanghai Television Team, which only blossomed but did not yield fruit most of the time, chief coach Ma Liangxing explained, "The National Team players just returned to the team three to four days ago, they didn't run-in well. The height of our forward players is not enough, the 'upper three paths' attack was often blocked. This is the first match of the Super League Matches this year. Some players are still nervous, they have pressure." Of course, he was satisfied with the whole manifestation of the team and the match result. Especially, he mentioned young player Zu Huihua.

The commander in chief of Dalian Team - Yan Qun, however, said humorously, "The Shanghai Television Team showed mercy, we only lost a ball. Certainly, we felt satisfactory."

Correspondent: Yang Renjue

Many different NEs and NERs can be observed in the review:

a) Named Entities:

Personal Name - 韩文霞 (Hang Wenxia[2]); 莫晨月 (Mo Chenyue); 马良行 (Ma Liangxing); 朱惠华 (Zu Huihua); 阎群 (Yan Qun); 杨仁杰 (Yang Renjue).

Team Name - 上视又利滋女足 (Shanghai Television You Li Zi Womens Football Team); 大连凯飞队 (Dalian Kai Fei Team); 上视队 (Shanghai Television Team); 大连队 (Dalian Team); 国家队 (National Team).

Identity - 门将 (goalkeeper); 前锋 (forward); 主教练 (chief coach); 球员 (player); 主帅 (commander in chief); 记者 (Correspondent).

[2] In Chinese personal names, the surname appears *before* the given name. Here, Hang is the surname; Wenxia is the given name.

Title - 全国女足超级联赛 (National Womens Football Super League Matches); ２０００年全国女足超级联赛 (2000 National Womens Football Super League Matches); 本年度女足超级联赛 (2000 WFSLM); 超级联赛 (Super League Matches).

Location name - 沪 (Hu, the abbreviation of the city Shanghai); 连 (Lian, the abbreviation of the city Dalian); 上海体育场 (Shanghai Stadium); 大连 (Dalian). Note that actually sometimes 沪, 连 and 大连 represent team names, such as 沪连之战 (the fight between Shanghai and Dalian.) and 上视队一球小胜大连 (Shanghai Television Team feebly won Dalian over Exactly one Ball). In Chapter 6, we will propose a strategy to identify such named entities without trigger words.

Time - ２０００年 (in 2000); 昨天下午 (yesterday afternoon); 前７０分钟 (former 70 minutes); 第７０分钟 (the 70th minute); 极大部分时间里 (at the most of time); 三四天 (three to four days).

b) Relationships between Named Entities

Person−Team - 韩文霞 (Hang Wenxia) − 国家队 (National Team); 莫晨月 (Mo Chenyue) − 上视队 (Shanghai Television Team); 马良行 (Ma Liangxing) − 上视队 (Shanghai Television Team); 阎群 (Yan Qun) − 大连队 (Dalian Team).

Person−Identity - 门将 (goalkeeper) − 韩文霞 (Hang Wenxia); 前锋 (forward) − 莫晨月 (Mo Chenyue); 球员 (player) − 朱惠华 (Zu Huihua); 主教练 (chief coach) − 马良行 (Ma Liangxing); 主帅 (commander in chief) − 阎群 (Yan Qun); 杨仁杰 (Yang Renjue) − 记者 (Correspondent).

Team−Identity - 国家队 (National Team) − 门将 (goalkeeper); 上视队 (Shanghai Television Team) − 前锋 (forward); 大连队 (Dalian Team) − 主帅 (commander in chief).

Team−Competition - 上视队 (Shanghai Television Team) − 全国女足超级联赛 (National Womens Football Super League Matches); 大连队 (Dalian Team) − 全国女足超级联赛 (National Woman Football Super League Matches); 上视又利滋女足 (Shanghai Television You Li Zi Womens Football Team) − ２０００年全国女足超级联赛 (2000 National Womens Football Super League Matches); 大连凯飞队 (Dalian Kai Fei Team) − ２０００年全国女足超级联赛 (2000 National Womens Football Super League Matches).

Hometeam−Visiting Team - 上视队 (Shanghai Television Team) − 大连队 (Dalian Team).

Winning team−Losing Team - 上视队 (Shanghai Television Team) − 大连队 (Dalian Team); 上视又利滋女足 (Shanghai Television You Li Zi Womens Football Team) − 大连凯飞队 (Dalian Kai Fei Team).

Competition−Time - 本年度女足超级联赛 (2000 WFSLM) − 昨天下午 (yesterday afternoon).

Competition−Location - 全国女足超级联赛 (National Womens Football Super League Matches) − 沪 (Hu); 本年度女足超级联赛 (2000 WFSLM) − 上海体育场 (Shanghai Stadium).

In addition, we can also see some dynamic descriptions in this text, such as 狂轰乱炸 (bombed savagely); 巍然不动 (stood firm); 一夫当关，万夫莫开 (one man can hold out against ten thousand); 地面进攻 (ground attack); 鱼跃头攻 (dove and attacked with somebody's head); 攻破 (broke through); 瓦解 (disintegrate) etc. Sometimes these verbs and phrases play important roles for identifying NERs. Therefore, the related domain verbs are collected in the domain ontology - *Sports Ontology* to express corresponding Movement concepts (see Chapter 5).

In our investigation, we defined 6 types of NEs and 14 kinds of NERs among them as the identified objects. The NEs include personal name, date or time, location name, team name, competition title and personal identity; while the NERs contain Person − Team, Person − Competition, Person − City / Province / Country, Person − Identification, Home Team − Visiting Team, Winning Team − Losing Team, Draw Team − Draw Team, Team − Competition, Team − City / Province / Country, Identification − Team, Competition − Date, Competition − Time, Competition − Location, Location − City / Province / Country.

1.3 Summary of the Main Results and Contributions

In this thesis, we have made efforts to improve the status of Chinese IE investigation that we mentioned in Section 1.1 and acquired the following scientific results:

- Adopting a *transformation-based error-driven* machine learning approach, we design and develop an error repairer in a specific domain. It can *simultaneously* correct the errors from word segmentation and POS tagging. In the experiments, we aimed at comparing the results of word segmentation and POS tagging with or without this component. The results have demonstrated that the average F-measure of word segmentation was enhanced by 5.11%; while that of POS tagging was even increased by 12.54%. Obviously, the quality of the baseline system has been distinctly improved.
- For *automatically* constructing FSC by NE recognition rules, we proposed an approach for that and developed a corresponding component. In this approach, the regular expressions employed in FSC are permitted to represent *complex constraints*. Thus, it extends the original definition of FSA (Finite State Automaton) and makes FSC more suitable to real-world applications. Additionally, the construction procedure of FSC is *transparent* for NE recognition rule developers. Therefore, FSC is more flexible and maintainable. On the basis of that, we further put forward a strategy for identifying NEs *without* trigger words, which cannot be identified by FSC. The experimental results have shown that total average recall, precision, and F-measure have achieved 83.38%, 82.79%, and 83.08% respectively.

- A learning and identification approach for NERs called positive and negative case-based learning and identification (PNCBL&I) is proposed. The learning in this approach is a supervised statistical learning. Actually, it is also a variant of memory-based learning. This approach pursues the improvement of the identification performance for NERs through *simultaneously* learning two *opposite* cases (NER and non-NER patterns), automatically selecting effective *multi-level* linguistic features from a predefined feature set for each NER and non-NER, and optimally making an identification tradeoff. The experimental results for this approach give the overall average recall, precision, and F-measure for 14 relations as 78.50%, 63.92% and 70.46% respectively. In addition, the above F-measure has been enhanced from 63.61% to 70.46% through adopting *both* positive *and* negative cases.

In summary, the main contributions in this thesis are briefly described as follows:

- A Chinese IE computational model has been proposed, which adopts a *hybrid* issue to improve the performance of word segmentation and POS tagging, to recognize NEs with or without trigger words, and to identifiy different NERs within a sentence or across sentences;
- An error repairer can simultaneously correct the errors of both word segmentation and POS tagging and remarkably improve the performance of the baseline system;
- A Sports Ontology with a hierarchical taxonomy has been suggested and developed. In addition to the relationships of the concepts in the taxonomy, the *relationships* between Movement and Object, Movement and Property, as well as Property and Object concepts have been also constructed;
- A novel approach for the automatic construction of FSC is proposed. In this approach, the regular expressions used to construct FSC are *allowed* to represent complex constraint symbols instead of atomic constraint ones. Thus, it extends the original definition of FSA, and makes FSC more suitable to *practical* applications;
- The strategy to identifying NEs without trigger words can give a solution for the *special* linguistic phenomena which can be *not* processed by FSC. It improves the performance of the NE identification;
- PNCBL&I, an innovative approach for NER learning and identifying is suggested, which pursues the improvement of the identification performance for NERs through learning two kinds of *contrary* cases (NER and non-NER patterns) simultaneously, automatically selecting *effective multi-level* linguistic features from a predefined feature set for each NER and non-NER, and making an *optimal* identification tradeoff;
- The prototype system CHINERIS, which adopts the above computational model, has been implemented in Java. The system can automatically identify 6 NEs and 14 NERs in the sports domain. Additionally, its run-time efficiency is acceptable and the system user interfaces are friendly as well.

1.4 Organization of the Thesis

This thesis consists of nine chapters and six appendixes. In this chapter, the investigation motivations and an overview of my thesis are introduced. Based on the introduction, a Chinese IE computational model is described, in which there are three stages to performing IE tasks, i.e., word processing, NE identification, and NER identification. In order to effectively fulfill these tasks, we adopt a *hybrid* issue - combining knowledge engineering and automatic trainable approaches in the model. For demonstrating the fruits of my thesis, such as *rationality* and *validity* of the approaches used in this model, the main results and contributions are listed in detail.

In Chapter 2, we will present an innovative language technology - Information Extraction (IE), including its history, major approaches, and significant systems. Especially, we will elaborate how this technology is applied to processing Chinese, that is, Chinese Information Extraction. In this aspect, we concentrate on analyzing the current research status. After that, some work related to the three stages in our model will be described.

Due to the considerable differences between Chinese and western languages, we will describe a number of *basic* conceptions concerning the Chinese language in Chapter 3. First of all, an outline with respect to Chinese is given. Then the evolution of Chinese is briefly presented. It is important that a survey of Chinese linguistics embodying morphology, grammar, and semantics with some explanatory examples is elaborated. Finally, a number of critical points regarding Chinese will be summarized in this chapter.

Chinese word processing is the task of the first stage in the IE computational model, whose *quality* is very important for consequent stages. In Chapter 4, a proposed approach that can improve the performance of Chinese word segmentation and part-of-speech (POS) tagging will be discussed. This approach employs transformation-based error-driven machine learning technique to develop an *error repairer* for correcting the errors from word segmentation and POS tagging of the baseline system.

For the sake of utilizing domain knowledge, an IE system may require an *ontology* providing the knowledge for extracting different objects such as NEs, NERs, and so on. A Sports Ontology whose primary concept expressions are introduced from *HowNet*, an bilingual (Chinese and English) on-line common-sense knowledge base, is described in Chapter 5, including its architecture and the definitions of concept descriptions. Moreover, this chapter further specifies how to use the information from Sports Ontology for identifying NEs without trigger words and computing semantic distance for identifying NERs.

Despite having different approaches to recognizing Chinese NE, considering Chinese NE's construction as well as the comprehensive factors such as *accuracy*, *efficiency*, and *robustness* of the NE identification, we utilize *Finite-State Cascades* (*FSC*) as a shallow parser to identify different NEs. Chapter 6 will state the basic conception of FSC and define recognition rules, the formal description for FSC, and its construction algorithm. In addition, it will describe the procedure for automatically constructing a recognizer by an example. After that, the procedure of the NE identification using FSC mechanism is also elaborated. Finally, in order to identify special constructions of NEs, that is, NEs *without* trigger words, we propose some *special* strategies for doing so.

In Chapter 7, we will suggest a learning and identifying approach for NERs - PNCBL&I. This approach pursues the improvement of the identification performance for NERs through learning two *opposite* cases (NER and non-NER patterns) simultaneously, automatically selecting *effective multi-level* linguistic features from a predefined feature set for each NER and non-NER, and achieving an *optimal* identification tradeoff. Based on the definitions and descriptions of the computational formula for this approach, we will further give experimental results and compare learning and identifying results with or without negative cases. Ultimately, we summarize the advantages of our approach and compare this approach with other related methods.

Chapter 8 goes into details for the system implementation, which contains system architecture, class definitions, as well as invoked relationships and executed sequence of methods in the classes for each core component.

The last chapter will give the conclusions of this thesis and the future work for the Chinese IE computational model.

Apart from the above nine chapters, we supply six appendixes to further explain some critical points of the chapters in the thesis. In Appendix A, some examples of Chinese morphology, grammar, and semantics will be provided. Appendix B illustrates a number of examples for error repairing rules used to correct Chinese word segmentation and POS tagging. A hierarchical taxonomy of Sports Ontology will be elaborated in Appendix C. Appendix D will give some examples for NE recognition rules. An example for the self-similarity calculation, feature selection, feature weight computing and identification threshold determination is explained in Appendix E. Finally, the system user interfaces are displayed by some screen shots in Appendix F.

Chapter 2

Information Extraction

2.1 Overview

Information Extraction (IE) aims to extract the *facts* from documents. Concretely speaking, IE extracts information from *actual* texts by computer at high speed, which are normally from publicly available electronic sources such as news wires, and maps it into predefined, structured representations (e.g., templates), which, when filled, represent an extraction of critical information from the original texts. Once extracted, the information can then be stored in databases to be queried, data mined, summarized in natural language, etc. (Gaizauskas et al., 1997).

It should emphasize that computational linguistic theories and technologies play a significant role in this emerging technology. Since the early 90's, IE technology has developed rapidly, driven by the series of *MUCs* (Grishman and Sundheim, 1996; Chinchor, 1998) in the government-sponsored TIPSTER program (Grishman, 1996). It is now coming onto the market and is of great significance for information end-user industries of all kinds, especially finance companies, banks, publishers and governments (Wilks, 1997). Therefore, IE is an important language technology with brilliant prospects.

Despite many important achievements in Chinese NLP (Cao, 2001), the investigation for Chinese IE is just in its *infancy* (Grishman et al., 1999; Wong et al., 1999). As mentioned in the previous chapter, only little work regarding Template Element (TE), Template Relation (TR), Coreference (CO), and Scenario Template (ST) IE tasks has been done. Even in the investigation of NE tasks, only a single approach has been adopted. Thereby, one of the motivations for our thesis research is to ameliorate the current research status of Chinese IE in some pivotal aspects. During our investigation, some related research work has been studied.

The structure in this chapter is arranged as follows. At first, the state of the art of IE is summarized, including the history, some approaches used for IE, and several significant systems of IE. Especially, the current research status of Chinese IE is presented. On the basis of that, we describe the research work related to our IE computational model in Section 2.3. Finally, we give a summary for this chapter in Section 2.4

2.2 State of the Art

2.2.1 History

First of all, the early work with respect to *AI Story Understanding* (Schank and Abelson, 1977) should be mentioned, because it may be one of the theoretical bases of IE. Schank and his students investigated some mechanisms such as *SAM* (*Script Applier Mechanism*) (Cullingford, 1978) and *PAM* (*Plan Applier Mechanism*) (Wilensky, 1978) etc. for Story Understanding. Their concentration, however, was less on language and more on the problems of representing and reasoning with the knowledge required for language. Using these ideas, a developed system called *FRUMP* (*Fast Reading Understanding and Memory Program*) (DeJong, 1979) can be viewed as an embryo of present IE systems. This system can recognize relevant information about seven events (earthquake, visit of state, terrorism event etc.), then fills the templates' slots with the information. Analogous work was accomplished at NYU (Sager, 1970) and other sites as well. But IE became a large-scale research effort in the late 1980's, principally because it was driven by a series of *DARPA* (*The Defense Advanced Research Projects Agency*) -sponsored evaluations, known as MUCs. Table 2.1 illustrates the staged time and evaluated corpus domain for each MUC. MUCs were promoted by DARPA in response to the situation that time, i.e. the enormous numbers of on-line texts. Being aware of the urgent need, DARPA initiated the field of IE, partly by focusing on a number of specific IE tasks.

MUC No.	Staged Time	Evaluated Corpus Domain
MUC-1	May, 1987	Naval Operation Reports
MUC-2	May, 1989	Naval Operation Reports
MUC-3	June, 1991	Terrorist Reports
MUC-4	May, 1992	Terrorist Reports
MUC-5	July, 1993	Joint-Ventures and Microelectronics
MUC-6	Sept., 1995	Management Succession
MUC-7	March, 1998	Rocket Payloads

Table 2.1 MUCs' Staged Times and Evaluated Corpus Domains

MUC-1 was an exploratory conference, each participant designed his own information format recorded in the document, and there was no official evaluation. In MUC-2, the task had been concentrated on *template* filling. The template consists of the *slots* such as the type of event, the agent, the time and place, the effect etc. The relevant information is extracted from events in the text. For MUC-2, the template has ten slots. Its corpus deals with military messages concerning naval sightings and engagement. By MUC-3, the task was moved to another domain - the reporting of terrorist events in Central and South America, and the template became *more* complex (18 slots). The same task was used for MUC-4, with a somewhat increased template complexity (24 slots). MUC-5, which was held as part of the TIPSTER program, opened up the evaluations for multiple tasks (international joint-ventures and electronic circuit fabrication) and languages (English and Japanese). In the aspect of the task complexity, the joint-venture task needed eleven templates with a total of 47 slots for the output - double the number of slots defined for MUC-4. In addition, a *nest* template structure was introduced in MUC-5, which can be recorded in a hierarchical structure. This is an innovation of MUC-5, about which the motivation is to gradually tend towards real-world applications. The primary goals of MUC-6 (Grishman and Sundheim, 1996) are (i) demonstrate task-independent component technologies of IE which should be useful at once; (ii) to promote portability in the IE tasks; (iii) to encourage "deeper understanding". There were sixteen participants in MUC-6. Of them, fifteen took part in the NE task, seven in CO, eleven in TE, and nine in ST. The highest performances[3] in MUC-6 can be seen in Table 2.2.

[3] In Table 1.2, P = Precision; R = Recall; F = F-measure with P and R weighted equally; E = English; C = Chinese; J = Japanese; S = Spanish; JV = Joint Venture; ME = Microelectronics

MUC-7 (Chinchor, 1998) is the last conference in MUCs. For the first time, the multilingual NE was evaluated using training and testing articles from comparable domains for all languages. The domain for training of all languages was airline crashes and that for testing of all languages was launch events. The TR was a newly defined task that identifies relationships between template elements. In MUC-7, TR was limited to relationships with organizations, i.e., employee_of, product_of, and location_of. The highest scores in MUC-7 and MET-2 (the Second Multilingual Entity Task) are also shown in Table 2.2.

The MUCs have directly yielded beneficial impacts for IE development: (i) assembling researchers to exchange their ideas, technologies, and systems; (ii) objectively evaluating and comparing IE technologies and systems; (iii) guiding further investigative directions; (iv) promoting IE innovative technologies and real-world applications. Although the MUCs are over, we are still confronted with new challenges from IE technologies (Grishman et al., 1999):

- more comprehensive processing of *linguistic* phenomena (e.g., aspectuals, reference, etc.);
- better modeling of *domain* knowledge including more common world knowledge;
- better automated *learning* methods to acquire background knowledge and to induce selection criteria for template slots.

In the next subsection, a number of state-of-the-art approaches used for IE will be described.

MUC No. / Task	**NE**	**CO**	**TE**	**TR**	**ST**	**ML**
MUC-3					R < 50% P < 70%	
MUC-4					F < 56%	
MUC-5					E JV F < 53% E ME F < 50%	J JV F < 64% J ME F < 57%
MUC-6	F < 97%	R < 63% P < 72%	F < 80%		F < 57%	
MUC-7	F < 94%	F < 62%	F < 87%	F < 76%	F < 51%	
MET No.						
MET-1	C F < 85% J F < 93% S F < 94%					
MET-2	C F < 91% J F < 87%					

Table 2.2 The Highest Performance Reported in MUC-3 through MUC-7 as well as MET-1 and MET-2

2.2.2 Major Approaches

Generally speaking, there exist four kinds of state-of-the-art approaches used for IE, i.e., knowledge engineering, automatically trainable, statistical, and hybrid approaches. Here, we would like to discuss these approaches with regard to mutual differences as well as their advantages and disadvantages.

The knowledge engineering approach mainly adopts linguistic knowledge to build grammatical and semantic rules for the components in IE systems. But this approach usually consumes a lot of manual work to establish refined rule libraries, because a high-performance system using this approach needs an iterative process in which the rules are tested and tuned again and again towards the expected goal. Therefore, sometimes this process is also called "hill climbing". In addition, domain knowledge has to be discovered by human experts in terms of observation of a corpus. One example for this approach is to utilize finite-state automata, and often *cascaded* automata (connected in serial) to break a complex problem into a sequence of easier subproblems. A typical system employing cascaded automata is *FASTUS* (*Finite State Automaton Text Understanding System*), whose synopsis can be seen in the next subsection. In short, the advantages of this approach are (i) the best performing systems for various information extraction tasks can be developed by hand; (ii) human ingenuity in establishing and tuning patterns is still in the lead.

The automatic training approach is quite different from the above approach. In this approach, different machine learning techniques are introduced to perform relevant tasks, e.g., *case-based leaning* (Aha et al., 1991), *memory-based learning* (Stanfill and Waltz, 1986; Daelemans, 1995; Daelemans et al., 2000), *explanation-based learning* (Nilsson, 1996) etc. In general, it is necessary that someone is knowledgeable regarding the domain and the task of selecting a corpus of texts and annotating the texts appropriately for the information being extracted. Once a suitable training corpus is annotated, a machine learning procedure can be started. Thus, the learning results will provide the system for analyzing new texts. The strong and weak points of this approach are not analogous to the knowledge engineering approach. Rather than focusing on producing rules, it focuses on producing training data. The corpus statistics or rules are automatically derived from the training data. Then they are applied to processing new data. So long as someone knowledgeable in the domain is available to annotate texts, the systems can be customized to a specific domain without interference from any developers. Except that, domain portability is considerable and "data driven" acquired rules enable all examples to be covered. As weaknesses, a great number of training data are often required for learning, but training data may be *difficult* or *expensive* to obtain. Moreover, changes to specifications may require reannotation of large quantities of training data.

In recent years, a number of statistical approaches have been applied to IE. e.g., *HMM* (*Hidden Markov Model*) (Viterbi, 1967; Baum et al., 1970), *VSM* (*Vector Space Model*) (Salton et al., 1975), and *Stochastic Grammar* (Booth, 1969; Booth and Thomson, 1973) etc. In general, the statistical approach depends on corpus analysis and statistics, which is an *empirical* approach. Using this approach, we can comprehend the complicated and extensive structures of language by specifying an appropriate general language model, and then inducing the values of parameters by applying statistical models to a large amount of language use. The advantages of this approach are (i) it can analyze and discover *fairly* fine distinctions of language phenomena; (ii) it can build a statistical model of *actual* language; (iii) because of (ii), it can resolve some *practical* problems of actual language texts. But it also has some disadvantages, such as it relies on statistical corpus including domain and distribution of language phenomena, etc., to a great extent; moreover, in general, it is effective only when its statistics derives from a lot of texts stored in corpus, but it may be *expensive*.

The fourth approach is a hybrid approach, which combines the above approaches giving play to their strong points. For example, combining knowledge engineering and automatic trainable approaches (e.g., a finite-state cascaded NE recognizer with rule learning and inference mechanism), or statistical learning (Duda and Hart, 1973) which ranges from simple calculation of averages to the construction of complex models such as Bayesian network (Wright, 1921; Wright, 1934; Good, 1961) and neural network (Cowan and Sharp, 1988a; 1988b). This approach has great impact since one approach can remedy the weak points of the other one, and is therefore very promising. In our investigation, we have also adopted this approach to establish a Chinese IE computational model (see Section 1.2).

2.2.3 Several Significant Systems

Since IE theories and technologies have been investigated, a number of significant IE systems are implemented and some of them were evaluated in the MUCs. Among them, the systems presented below have considerable in influence:

a) DIDEROT

Under the TIPSTER Project, this system (Cowie et al., 1993) was developed at CRL[4] and Brandeis University. DIDEROT can be applied to two application domains, i.e., business and micro-electronics. At the same time, the IE tasks can be carried out for texts in two languages (English and Japanese). The three innovative aspects involved in the system are (i) the use of multi-pass finite-state feature tagging for parsing NEs, e.g., dates, locations, person, and company names, in both languages; (ii) the automatic construction of the core lexicon for a particular language, and the use of statistical methods applied to corpus to build a specific domain lexicon; (iii) the automatic conversion to DCG (Definite Clause Grammar) (Pereira and Warren, 1980) parse rules of lexical co-specification information, tuned against the corpus for each lexical item. The system architecture has of five components - merger (semantic and POS tags), compound noun recognizer, parser, reference resolver, and template. The performances (recall and precision) of the NE identification for English and Japanese texts are (13% and 55%) as well as (22% and 60%) respectively.

b) LaSIE (Large Scale Information Extraction)

LaSIE (Gaizauskas et al., 1995) is implemented by the University of Sheffield. It is an *integrated* system for natural language engineering and can also serve as IE. This system is a pipelined architecture consisting of three main components - lexical preprocessing, parsing plus semantic interpretation, and discourse interpretation. The primary characters of the system are (i) due to the adoption of the integrated approach, it allows knowledge at *different* linguistic levels to be applied to each task defined by MUC-6; (ii) lexical information for parsing is *dynamically* calculated through part-of-speech-tagging and morphological analysis; (iii) the grammar used is semi-automatically derived from the Penn TreeBank corpus; (iv) a world model in the ontology is acquired and used for the CO and ST tasks; (v) contains a summarization model for a brief natural language summary of scenario events. LaSIE can identify NEs including location, personal titles, organizations, dates/times, and currencies. Moreover, it can perform all four of the MUC-6 tasks, i.e., NE, CO, TE, ST. The following table shows the performance of LaSIE for the above four tasks in MUC-6.

[4] Computer Research Laboratory, New Mexico State University.

Task Type	**NE**			**CO**		**TE**			**ST**		
Performance Type	**R**	**P**	**P&R**[5]	**R**	**P**	**R**	**P**	**P&R**	**R**	**P**	**P&R**
Performance	84	94	89.06	0.51	0.71	66	74	69.80	37	73	48.96

Table 2.2 Performance of LaSIE for the Four Tasks in MUC-6

LaSIE-II (Humphreys et al., 1998) has *significant* differences from LaSIE. From the viewpoint of the architecture, one of the most important developments in LaSIE-II is its modularisation and integration into the GATE platform (see below). It was developed using GATE, which manages all the information about the texts that are produced by each module, and provides graphical tools for visualising the information, selecting control flow by different module combinations and running the IE system over sets of texts. Overall, LaSIE-II primarily includes 9 modules, i.e., tokenizer, gazetteer lookup, sentence splitter, brill tagger, tagged morph, buchart parser, name matcher, discourse interpreter, template writer.

In the lexical preprocessing, the "gazetteer lookup" module is arranged from immediately before the parser in LaSIE to immediately after the tokenizer in LaSIE-II. Thus, it is able to improve the accuracy of the "sentence splitter" module. In addition, whereas originally only particular tags were matched (nouns, adjectives, determiners, conjunctions, numerals, symbols), the "gazetteer lookup" tries to match *all* tokens, so that it no longer suffers from tagging errors. In order to adapt *different* domains, a top-level configuration file defines a set of plain text lists and type (subtype) values to be assigned to matches in each list. This module can be switched between domains by specifying *alternative* configuration files.

In the parsing, the primary changes are a remarkably improved grammar development environment, and a completely rewritten and extended grammar. The module "buchart parser" can be run under the GATE graphical interface. After each grammar is run, its results may be viewed using a tree viewer, and if modifications are required, the grammar may be edited and rerun without leaving GATE for any recompilation process. Therefore, the benefits with the new graphical tools are to allow more rapid development and verification of subgrammars, and support grammar development by different persons working on *different* subgrammars. On the other hand, the phrasal grammars were completely rewritten and compartmentalised, using a combination of general principles - a university grammar of English and iterative refinement employing the MUC-7 training data.

In the discourse interpretation, except for some gazetteer lists and the related grammar rules, all domain specific knowledge is gathered in the domain model of the discourse interpreter. In LaSIE, this model is expressed using a semantic net. In LaSIE-II, however, the initial domain model was constructed directly from the template definition of the MUC-7. During processing, the instances and properties from the semantic representation of a text, i.e., quasi-logical form (QLF) produced by the parser are added to the domain model. The QLF of each sentence is processed in the various components (add semantics, add presuppositions, object coreference, add consequences, and event coreference), gradually specialising the domain model to evolve a discourse model. Its knowledge is then passed to the template writer.

[5] P&R means P and R are equally weighted.

LaSIE-II has fulfilled all five tasks of the MUC-7, among which TR (Template Relation) is a new task. It was the only system to take part in all of the MUC-7 tasks. Table 2.3 illustrates the performance of LaSIE-II for the above tasks in MUC-7.

Task Type	**NE**			**CO**			**TE**		
Performance Type	**R**	**P**	**P&R**	**R**	**P**	**P&R**	**R**	**P**	**P&R**
Performance	83	89	85.83	56.10	68.80	61.80	75	80	77.17

Task Type	**TR**			**ST**		
Performance Type	**R**	**P**	**P&R**	**R**	**P**	**P&R**
Performance	41	82	54.70	47	42	44.04

Table 2.3 Performance of LaSIE-II for the Five Tasks in MUC-7

c) FASTUS (Finite State Automaton Text Understanding System)

FASTUS (Hobbs et al., 1996) was designed and developed by SRI[6] in 1992. Originally, it was considered to be a preprocessor for TACITUS (The Abductive Commonsense Inference Text Understanding System) (Hobbs et al., 1991), which is an interpreting system for natural language texts. The crucial idea in FASTUS is the "*cascade*" components in "cascaded, nondeterministic finite-state automata". The earlier components identify smaller linguistic constituents in a largely domain-independent mode. They utilize *purely* linguistic knowledge to process the portion of the syntactic structures of sentences which linguistic approaches can reliably determine and the system only needs little modification from domain to domain. On the other hand, the later components receive these linguistic constituents as input and seek *domain-dependent* patterns among them.

There are five components in the overall architecture, which in turn process complex words (e.g., multi-words, company names, personal names, locations, dates, times and other entities), basic phrases, complex phrases, domain events, and merging structures. Thus, phrases can be identified *reliably* with purely syntactic information. Moreover, they provide *precisely* the elements required for specifying the event patterns. Constructing a system in such a way gives it *greater* portability among domains and allows it to *easily* acquire new patterns.

The advantages of the system are: (i) it has simple conceptions, that is, a set of cascaded finite-state automata; (ii) it is effective, in other words, it has been among the leaders in past evaluations; (iii) the run-time of the system is very fast; (iv) because the system provides a direct link between the texts being analysed and the data being extracted, it can develop the system in a very short time.

SRI International took part in the evaluation for each of the MUC-6 tasks (Appelt et al., 1995) using the latest version of FASTUS system (Appelt et al., 1993). In Table 2.4, the performance of FASTUS for the four tasks in MUC-6 is listed.

[6] Artificial Intelligence Center, SRI International, Menlo Park, California.

Task Type	NE			CO		TE			ST		
Performance Type	**R**	**P**	**P&R**	**R**	**P**	**R**	**P**	**P&R**	**R**	**P**	**P&R**
Performance	92	96	94	59	72	74	76	75	44	61	51

Table 2.4 Performance of FASTUS for the Four Tasks in MUC-6

d) GATE (General Architecture for Text Engineering)

GATE (Cunningham et al., 1997) is an architecture, development environment and framework for establishing systems that process natural languages. It has been in development at the University of Sheffield since 1995, and has been utilized for many R&D projects, including IE in multiple languages and IE for multiple tasks and clients (Bontcheva et al., 2003). The development of the second version of GATE started in 1999 and led to a complete redesign of the system which is implemented using pure Java codes. Additionally, it can fully support Unicode data, allowing the users to open, visualize and process documents in different languages.

There are three primary types of components in GATE architecture: (i) language resources (LRs), which store some kinds of linguistic data such as documents, corpora, ontologies and provide services for accessing them; (ii) processing resources (PRs), which are resources whose character is principally programmatic or algorithmic, e.g., a POS tagger or a parser. In most cases, PRs are utilized to process the data supplied by one or more LRs; (iii) visual resources (VRs), which are graphical components displayed by the user interface and allowing the visualization and editing of other types of resources or the control of the execution flow.

Provided with GATE is a set of reusable PRs for general NLP tasks. PRs are integrated to build ANNIE (A Nearly-New IE system), which primarily consists of five resources: tokeniser, sentence splitter, POS tagger, gazetteer, semantic tagger, and orthomatcher. Among them, the semantic tagger is composed of hand-crafted rules written in the JAPE (Java Annotations Pattern Engine) language, which describes patterns to be matched and annotations to be created. A JAPE grammar consists of a set of phases, each of which is composed of a set of patterns / action rules, and which run sequentially. The orthomatcher's main objective is to perform coreference processing, or entity tracking, by identifying relations between entities. It can also improve NE identification by annotating previously unclassified names, based on relations with existing entities.

Table 2.5 demonstrates the performance of ANNIE for English NEs' identification.

Entity Type	Address	Date	Location	Money	Organization	Percent	Person	Overall
P	81	67	88	82	75	100	68	82
R	81	77	96	47	39	82	78	67
P&R	81	71.65	91.83	59.75	51.32	90.11	72.66	73.74

Table 2.5 Performance of ANNIE for English Named Entities' Identification

2.2.4 Chinese Information Extraction

The investigation of Chinese IE began in the early 90's. National Taiwan University started to study Chinese NE extraction in 1993. Chen et al. (1998) adopted different types of information from different levels of text to extract NEs, including character conditions, statistic information, titles, punctuation marks, organization and location trigger words, speech-act and locative verbs, cache and n-gram model. Their *NTU System* was evaluated in MET-2, whose F-measure P&R was 79.61%. Analogously, *Kent Ridge Digital Labs System* (Yu et al., 1998) from Singapore was also evaluated in MET-2. This system is based on the statistical approach, so its language model is built by training corpus. During the construction of the language model, domain knowledge is integrated in the model in a systematic and generic way. In the system architecture, core components embody sentence segmentor and tokenizer, text analyzer, hypothesis generator, and disambiguation module. The first component accepts a stream of characters as input and transforms it into a sequence of sentences and tokens; the second one provides the analysis necessary for the particular task, e.g., an orthographic, syntactic, or semantic task; the third one determines the possible boundary and category for each word or token, depending on named entities' prefixes, suffixes, trigger words and local context information. Time, date, money, and percentage are extracted by pattern-matching rules; the fourth one adopted the Viterbi algorithm (Viterbi, 1967) to compute the probability with which a tag sequence corresponds to a word sequence. The F-measure P&R of the system achieved 86.38%, which was better than the NTU system.

In addition to the above work, other researchers have done a great deal of valuable work in this field using *linguistic*, *statistical*, *machine learning*, *language model* etc. approaches over the past few years as well. Some examples are listed as follows:

In (Zhu and Yao, 1998), the authors proposed a five-stage Chinese IE model. It is a pipeline architecture that consists of filter, nominal phrase, sentence, paragraph and template stages. The filter stage leaches irrelevant texts from input texts; the nominal phrase stage recognizes different entities in the text and builds corresponding entity templates; the sentence stage realizes the construction of event templates and builds the relationships of entities; the paragraph stage settles coreference resolutions between sentences and constructs the entity-event network; finally, the template stage formats the analysis results and outputs them into a predefined template.

Wong et al. (1999) suggested an automatic verb classified approach for Chinese temporal IE. Provided with a Chinese verb dictionary, verbs are first classified into four classes, namely, instantaneous verbs, activity verbs, static verbs, and ambiguity verbs. Then, their corresponding types of temporal concepts can be determined, i.e., static, durative, or telic type, according to the surrounding contexts.

A knowledge extraction process to extract the knowledge for identifying Chinese organization names was proposed by Chen and Chen (2000). This approach utilizes the structure property, statistical property and partial linguistic knowledge of the organization names to identify new organizations from domain texts. With a high standard of threshold values, new organization names can be identified with very high precision.

In (Zhang and Zhou, 2000), the authors presented a *memory-based learning* approach for identifying Chinese NEs (personal and organization name) and their relations (*employee-of*, *product-of*, and *location-of*). Concretely speaking, they employed memory-based learning to classify named entities and relations. The preliminary experiments showed that the performance of the system is comparable to or better than other existing trainable methods.

A statistical language model (LM) was used to identify NEs (Sun et al., 2002). In this work, word segmentation and NE identification are integrated into a unified framework that consists of several *class-based* language models. On the other hand, a hierarchical structure for one of the LMs is employed, so that the nested entities in organization names can be identified. In the evaluation, the *integrated heuristic* information improves the recall and precision of the system. The cache-based LM increases the recall of NE identification to some extent. Except that, some rules associated with abbreviations of NEs had remarkably increased the performance.

Ye et al. (2002) thought that the *uncertainty* in word segmentation and *flexibility* of linguistic structure are two critical problems in Chinese NE identification. They used a rationality model in a *multi-agent* framework to solve these problems. In order to evaluate and detect all possible NEs in a text, they adopted a greedy strategy and utilized the NE rationality measures. Then the agent-based reasoning and negotiation was applied to select the best possible NEs. The experimental results showed the system is robust and is able to handle different NE models. Moreover, the F-measure on the test for the MET-2 corpus achieved over 92% on *all* NE types.

Single character named entity (*SCNE*) is a kind of NEs, which is composed of one Chinese character. SCNE is a common linguistic phenomenon in written Chinese text. Zhu et al. (2003) utilized the *improved source-channel model* (Gao et al., 2003) to identify single character location (SCL) names and single character person (SCP) names. The experimental results of this approach are the F-measure of 81.01% for SCL and that of 68.02% for SCP, which are better than those of the maximum entropy (ME) and vector space model (VSM) methods.

2.3 Related Work

In this section, some of the main research work related to our IE computational model, which is described in Section 1.2, is described, which is associated with three stages, namely, word segmentation and POS tagging, NE identification, and NER identification.

2.3.1 Word Segmentation and Part-of-Speech Tagging

The baseline system of word segmentation and POS tagging in our system was developed by Shanxi University, China (Liu, 2000; Liu, 2001). In the system, the word segmentation component is principally based on the AB (Association-Backtracking) algorithm, which relies not only on word libraries, but also makes use of other linguistic knowledge such as word-building, form-building and syntactical knowledge. It adopts practical word segmentation rules to solve ambiguous structure problems for improving the efficiency and the precision of word segmentation. Therefore, this algorithm is not simple word matching, but the combination of association (used for particle formation) and backtracking (used for disambiguity). In addition, the component of POS tagging uses the probability statistic model, CLAWS, VOLSUNGA (DeRose, 1988; Liu, 2000), and the corresponding transmutation algorithms to tag different POSs. In the system, the POS tag set includes 25 categories, 85 subcategories and 64 punctuations.

Brill (1995) proposed a *transformation-based* approach for corpus-based learning. Concretely, it was applied to POS tagging and obtained competitive performance results compared with stochastic taggers on tagging both unknown and known words. This learning approach had also been applied to a number of other NLP tasks, including preposition phrase attachment disambiguation (Brill and Resnik, 1994), bracketing text (Brill, 1993a), and labeling non-terminal nodes (Brill, 1993b).

Palmer (1997) applied Brill's transformation-based error-driven learning to word segmentation problems. The models which are yielded are both more compact and easier to interpret than stochastic models. On the other hand, their performances are comparable to that of the best hand-crafted systems. Hockenmaier and Brew (1998) extended Palmer's work by showing some effects of variation in corpus *size* and rule *complexity*. The experiments performed satisfactorily even with a very simple initial state tagger. A major reason depends on the large size of the training corpus which were employed.

2.3.2 Named Entity Identification

Finite-State Cascades (FSC) are a finite-state technique used for NLP tasks. During the development of CASS (Cascades Analysis of Syntactic Structure) System, Abney (1990) has applied the embryo of this technique to it. CASS consists of three core filters, that is, the chunk, clause, and parse filters. The parser is a pipeline architecture. Each filter makes a definite decision regarding a specific problem, such as POS disambiguation or identification for the subject and predicate of simplex clauses. The experiment has proved that the parser is considerably *fast* and *reliable*. Further, Abney (1996) used this approach to parse unrestricted English and German texts. Deterministic parsers constructed by FSC are fast and reliable. In fact, they are *more* accurate than exhaustive-search stochastic context-free parsers.

Moreover, the FSC technique has been applied to other IE systems, shallow text processing systems, and even development environments for text engineering as well, e.g., FASTUS (Finite State Automaton Text Understanding System) (Appelt et al., 1993; Hobbs et al., 1996), GATE (General Architecture for Text Engineering) (Cunningham et al., 1997), CCSP (Cascaded Chinese Syntactic Parser) (Zhang, 1998) and SproUT (Shallow Processing with Unification and Typed Feature Structures) (Becker et al., 2002) etc. Among them, CCSP system as a shallow parser can analyze different Chinese phrases and sentence structures and SproUT system can compile regular expressions of Chinese grammar and produce relevant automata.

Chen et al. (1998) utilized various information from the linguistic levels of texts to identify different NEs, which embody character conditions, statistic information, titles, punctuation marks, organization and location trigger words, speech-act and locative verbs, cache and n-gram model.

HowNet (Dong and Dong, 2000) is a bilingual (Chinese and English) on-line common-sense knowledgebase depicting *inter-conceptual* relations and *inter-attribute* relations of concepts. The knowledge relationships constructed by HowNet form a graph rather than a tree structure. It is devoted to describing the general and specific properties of concepts. In recent years, a number of researchers have employed HowNet for their research work and applications: Zhou and Feng (2000) established three data tables, i.e., concept, feature, and relation tables, to construct the bi-directions and multi-angles connections among them, as well as integrated all the information in HowNet into a relational network. For further study of Chinese information retrieval and knowledge reasoning, this approach provides an effective means of knowledge acquisition from HowNet. Additionally, Liu and Li (2002) suggested an approach to computing the word similarity based on HowNet.

2.3.3 Named Entity Relation Identification

Unlike most learning algorithms, case-based, also called exemplar-based or instance-based, approaches do not construct an abstract hypothesis but instead base classification of test instance on similarity to specific training cases, e.g., (Aha et al., 1991). The distance between a test instance and every training instance can be calculated for deciding instance classification, e.g., k-nearest neighbor or k-NN approach (Cover and Hart, 1967; Duda and Hart, 1973).

In the feature relevance, *wrapper* methods (John et al., 1994) for feature selection yield a set of candidate features, execute the corresponding induction algorithm with these features, and then use the accuracy of the resulting concept description to evaluate the feature set.

Cardie (1996) assigned *weights* to features based on linguistic or cognitive preferences. There are three biases in her approach, i.e., the *recency* bias (assigning higher weights to features that represent temporally recent information), the *restricted memory* bias (keeping the n features with the highest weights), and the *focus of attention* bias (assigning higher weights to features that correspond to words or constituents in focus, e.g., the subject of a sentence).

Memory-based learning (*MBL*) methods (Stanfill and Waltz, 1986; Daelemans, 1995; Daelemans et al., 2000) are supervised learning methods that are to use similarity metrics in the instance space for supporting future predictions. The idea behind the methods is that each language experience leaves a memory trace used to instruct future processing. When a prediction is to be made with regard to a new instance, relevant instances are chosen from memory, and the predication is made by analogy to these. Technically, MBL methods use variations of the classic k-NN algorithm. Instances are stored in memory together with the associated label. When a new instance is processed, the k nearest neighbors of this target instance are retrieved from memory, according to some metric on the instance space and the target instance is classified by extrapolating on the labels of these k neighbors in some way.

Zhang and Zhou (2000) proposed a trainable method for extracting Chinese NEs and NERs. They viewed the entire problem as a series of classification problems and employed MBL to resolve them. The authors claimed that the system performance is proved comparable to or better than other existing trainable methods, such as HMM and CRYSTAL, by the preliminary experiment.

2.4 Summary

In this chapter, we have introduced an innovative language technology - Information Extraction, including its history, state-of-the-art approaches, and significant systems. In particular, we described the current research status on Chinese IE. Apart from that, for the sake of specifying our work basis, some work related to our computational model has also been presented.

Since the late 1980’s, a series of MUCs have made IE theories and techniques more and more mature. In general, there exist four kinds of state-of-the-art approaches used for IE, that is, *knowledge engineering*, *automatically trainable*, *statistical* and *hybrid* approaches. The knowledge engineering approach principally uses linguistic knowledge to establish grammatical and semantic rules for the components in IE systems. But its main shortcoming is that it requires much manual work to build rule libraries. In contrast to the above approach, the automatic training approach adopts machine learning techniques for training annotated corpus, so that the learning results can be used for analyzing new texts. However, a great of number training data are usually required for learning, it is difficult to supply or expensive to obtain them. The statistical approaches depend on corpus analysis and statistics, thereby they analyze structures of language by specifying a general language model and induce the values of parameters by applying statistical models. Because they rely on statistical corpus to a great extent and need a great number of texts in corpus, it may be unstable and expensive. The last approach exploits the former three approaches' advantages to the full, making this approach a promising one.

Furthermore, there have been significant IE systems over the past ten years. DIDEROT can be applied to two application domains, i.e., business and micro-electronics, and two language texts, namely, English and Japanese. LaSIE and LaSIE-II are integrated systems for natural language engineering, which can serve as IE tasks. One of the most primary developments in LaSIE-II is its modularization and integration into the GATE platform, which is an architecture, development environment and framework for establishing systems that process natural language. LaSIE-II was developed under the GATE environment, and provides graphical tools for visualizing the information. The key design idea of FASTUS is the "cascade" components in Finite State Cascades (FSC). The earlier components identify smaller linguistic constituents in a largely domain-independent mode. They utilize purely linguistic knowledge to process the part of the syntactical structure of sentences, which linguistic methods can reliably fix and the system only needs little modification from domain to domain. On the other hand, the later components get these linguistic constituents as input and find domain-dependent patterns among them.

As an investigation for IE technology applied to Chinese texts, it began in the early 90's. Two systems (NTU and KRDL) participated in the evaluation of MET-2. The first system adopted different types of information from different levels of text to identify NEs. The second one is based on the statistical approach, so its language model is established by training corpus. In addition to the above work, other research projects have also done a lot of meritorious work in this field using linguistic, statistical, machine learning, language models etc., such as a five-stage Chinese IE model, an automatic verb classified approach for Chinese temporal IE, a knowledge extraction process to extract the knowledge for identifying Chinese organization names, a memory-based learning approach for identifying Chinese NEs and NERs, a statistical language model (LM) used to identify NEs, a rationality model in a multi-agent framework for identifying Chinese NEs, the identification for single character NEs and so on.

We divided related work into three aspects corresponding to three stages in our Chinese IE computational model. The related work in the first stage chiefly deals with the baseline system of word segmentation and POS tagging and the transformation-based error-driven machine learning method used in our system. The second part of related work includes the FSC technique, NTU system, and HowNet knowledge base, etc. The related work associated with the last stage is case-based machine learning, wrapper methods for feature selection, automatic weight assignment approach and memory-based learning.

In order to explain the differences between the Chinese language and western languages, we will elaborate a number of basic conceptions concerning the Chinese language along with various examples in the next chapter.

Chapter 3

Chinese Language

Since Chinese language has a very different topology in comparing to western languages, I would like to give a fundamental introduction to Chinese language, before stating my major research work in details. First of all, I will outline the Chinese language in the next section. Then the evolution of the Chinese language, including several different varieties of the language, will be presented in Section 3.2. Using a large number of examples, Section 3.3 makes a survey of Chinese linguistics encompassing morphology, grammar, and semantics. Finally, a summary description with respect to the Chinese language will be provided in Section 3.4.

3.1 Overview

Chinese language is the official language of the Chinese Mainland and Taiwan. Beginning with the Xia Dynasty (21st - 17th century BC), which was the first dynasty of China, Chinese was formed, used and developed. The process of Chinese development can be divided into three historical periods, that is, Classical Chinese, Pre-modern Chinese and Modern Chinese respectively (Lü, 1985). Among China's more than 1 billion people (over 980 million in Mainland and 19 million in Taiwan), approximately 95 percent of them speak Chinese, as opposed to the non-Chinese languages—such as Tibetan, Mongolian, Lolo, Miao, and Tai—spoken by minorities (Denison, 2003). A great number of emigrant people who speak Chinese can be found throughout the whole of Southeast Asia, especially in Hong Kong, Indonesia, Malaysia, Thailand, and Singapore. In addition, some Chinese-speaking communities can be found in many other parts of the world, especially in the USA.

Chinese belongs to the family of Sino-Tibetan languages. It has a large character set that involves more than 48,000 characters, which have been collected by (The Editorial Office of Zhonghua Book Company, 1915). Besides a core vocabulary and sounds, Chinese has primary features that distinguish it from most Western languages:

- It is not a segmented language, which means there are no boundary markers between words;
- It is a monosyllabic language. In general, every morpheme is monosyllabic and corresponds to a Chinese character;
- Chinese words have even fewer inflections than the words of western languages such as English or German words. The word order and particle[7] play an important role in Chinese grammar;

[7] Here it includes adverbs, conjunctions, pronouns, propositions, quantifiers, auxiliary words, interjections, etc.

- It is tonal. In order to indicate differences in meaning between words that are similar in sound, tone languages assign a distinctive relative pitch-high or low-or a distinctive pitch contour-level, rising, or falling to words.

Additionally, there are a number of characters in the Chinese language which deserve mention: (i) Chinese word formation is identical with its sentence construction; (ii) Chinese compound words and basic phrases are composed of meaningful morphemes; (iii) Compared to western languages, Chinese sentence structure is more flexible and sentence elements are more loosely-coupled; (iv) Some constituents in Chinese sentences can be omitted in certain contexts. The detailed explanations concerning these characters will be given in Section 3.3.

3.2 Evolution of Chinese Language

The 20th-century movement for language reform in China brought the most ambitious program of language normalization in the world (Halsall, 2003). This program has three aims: (i) to *simplify* the characters of classical written Chinese, by cutting down on their number, and reducing the number of strokes it takes to write a character; (ii) to provide a *single* means of spoken communication throughout the whole of China, by popularizing the Beijing-based variety, which has been chosen as a standard; (iii) to introduce a *phonetic alphabet*, which would gradually replace the Chinese characters in daily use.

There have been actions to reform the language from as early as the 2nd century BC, but none of these previous movements can be comparable in the sense of complexity to the present-day program, in which frequent reference is made to the names of several different varieties of the Chinese language:

- *文言文* (*literary speech* or *body of classical writing*). The refined literary language, recorded from around 1,500 BC and the traditional unified medium for all varieties of Chinese. It differs greatly from everyday speech, especially in its *terse* grammatical style and *specialized* literary vocabulary. It is now less widely used, because of the success of the current reform movement of written Chinese.
- *白话文* (*colloquial language*). A *simplified*, *vernacular* style of writing, introduced by the literary reformer Hu Shi in 1917, to make the language more widely known to the public, and to permit the expression of new ideas. A method of writing which encodes everyday speech developed as early as the Song Dynasty (AD 960-1279), but made little impact on the dominant literary speech. However, the 'May Fourth Movement' (which originated in political demonstrations on 4 May, 1919 after the Paris Peace Conference) adopted Hu's ideas, and colloquial language was recognized as the national language in 1922.
- *普通话* (*common language*). The variety chosen as a *standard* for the whole of China, and widely promulgated under this name after the establishment of the People's Republic of China in 1949. (In Taiwan, it goes under the name of 国语 (national speech). In the West, it is generally referred to simply as 'Mandarin'.) It takes the pronunciation of Beijing colloquialism, the grammar of the spoken Chinese, and the vocabulary of colloquial Chinese literature. In 1956, it became the medium of instruction in all schools, and a policy of

promoting its use began. It is now the most widely used form of spoken Chinese, and is the standard written medium for almost all kinds of publication.

- *拼音* (*phonetic spelling*). After several previous attempts to write Chinese using the letters of the Roman alphabet, this 58-symbol writing system was finally adopted by authorities (Chinese Language Reform Commission of China, 1958). Its main aims are to facilitate the spread of the common language, and the learning of Chinese characters. Phonetic spelling is now in widespread use. In the 1970s, for example, a list of standard spellings of Chinese place names was compiled, and a new map of China was published using the alphabet. New codes were devised for various uses such as telegraphy, flag signals, braille, and deaf finger-spelling.

The future of the reform program is not entirely clear. It is possible that phonetic spelling will ultimately supplant the general use of characters, or there might a movement for preserving the traditional written language. With common language, new varieties of regional pronunciation are certain to develop, which may lead to understanding problems. And if common language continues as a popular means of communication, it might be necessary to consider the potential conflict with regional dialects (for example, whether local words should be used). Much will depend on how flexibly the authorities interpret the notion of standard, and whether they are able to achieve a balance between the competing pressures of respecting popular usage (where there is a strong case for variety) and the need for national communication (for which some linguistic rules might have to be established).

3.3 A Survey of Chinese Linguistics

Chinese linguistics embodies enriched contents in morphology, grammar, and semantics. The motivation for providing this survey here is that we want to highlight some fundamental concepts which reflect the essence of Chinese linguistics. On the other hand, these concepts are associated with our research work as well. In the following subsections, in accordance with a number of prevalent works regarding Chinese linguistics (Chao, 1968; Li and Thompson, 1981; Qian et al., 1995; Leech, 1987; Jia, 1999; Wu, 1999), firstly, the morphology of Chinese, which is rather different from western languages, is described. Then Chinese grammar that mainly covers part-of-speech, different phrases, and sentence patterns etc. will be introduced. Finally, Chinese semantics, especially lexical and sentence[8] semantics, will be elaborated. Note that in order to save space, we transfer many explanatory examples in regard to Chinese morphology, grammar, and semantics from this section to Appendix A.

3.3.1 Morphology

Morphology deals with the *internal structure* of words, which is described in terms of morpheme, that is, the smallest meaningful element in language (Li and Thompson, 1981). In general, Chinese phonology divides the syllable into an *initial* and a *final*. Additionally, each syllable has a *tone* (total four tones), which is primarily the pitch pattern of the voiced part of the syllable. If the initial is voiced, the tone begins with the initial and spreads over the whole syllable, while, if the initial is voiceless, the tone is spread over the final only. As to the relationship between morpheme and character, generally speaking, the *monosyllabic*

[8] Sentence semantics means the semantics at the sentence level.

morpheme is the basic form of morphemes and corresponds to a determined character in the written language. A large majority of Chinese characters have aone-to-one relationship with morphemes. On the other hand, a Chinese syllabic unit (with a specific tone) represents several morphemes, that is to say, it has a one-to-many relationship with the characters that are called *homonymous characters*. In contrast, multiple tones may be pronounced in some morphemes, namely, *polyphonic characters*.

The relationships between a character and a word can be listed as below (Wang and Zhou et al., 1999):

- A character represents a word, e.g., 人 (human being), 山 (mountain): each word corresponds to one monosyllabic morpheme.
- More characters denote a word, e.g., 徘徊 (pace up and down), 巧克力 (chocolate): they consist of *polysyllabic morphemes*; 图书馆 (library): they are composed of three morphemes.
- A character expresses more word senses, e.g., 花 (flower; coloured; spend) may represent different meaning in words, such as 玫瑰花 (rose), 花衣服 (bright-coloured clothes) or 花钱 (spend money).
- Different characters have the same word meaning, e.g., the simplified Chinese character 汉 (the Han Dynasty or Chinese) and corresponding original complex character 漢 express the same word.

The monosyllabic words come from the single-morpheme character set. They are called *simple words*, which represent natural phenomena and things; product and life material; human body organs; basic movement, action, properties and status; or time, direction, quantity and reference. These words have *universality* (are universally used), *stability* (have a long-history) and *combinability* (are capable of combining with other words to form new words).

Most of Chinese words are disyllabic words. They have two types: *simple word* and *compound word*.

- *Simple word*: In general, there are four types of simple words.
  - *Unbroken word*. It consists of two characters, often alliterated or rhymed.
  - *Phone-reduplicate word*. It must be composed of two same-syllabic morphemes. When the reduplicate morpheme is monosyllabic, the second syllable takes a neutral tone.
  - *Transliterated word*. It is translated from the word of another language (e.g., English) in terms of its pronunciation. The characters in the word have no meaning, only identify the meaning by their pronunciation.
  - *Onomatopoetic word*.

  (See Examples A.1.1 – A.1.4.)
- *Compound word*: It consists of two or more than two morphemes that have the correlation of meaning. Depending on the meaning and the position of the morphemes in the compound words, we can divide them into word roots and

affixes. The word root has a practical meaning and can occur in different positions of a compound word; while the affix is a bound morpheme that is added to other morphemes to form larger unit. Often, it is a grammatical morpheme indicating number, aspect, and so on. Of the three types of affixes - *prefixes*, *suffixes*, and *infixes* - prefixes and infixes are extremely *rare* in Chinese, while suffixes are slightly more numerous (Chao, 1968).

(See Examples A.1.5 – A.1.7.)

Generally speaking, there are two types of the construction mode for compound words. One is the *complex* form that is composed of word roots. Another is the *adjunctive* form that consists of word roots, prefixes and suffixes.

a) Complex form

a. *Coordination*

The meaning of the first morpheme is the same as, close to or opposite from the meaning of the second morpheme.

b. *Verb-Object*

In such a word, according to the meaning, the preceding morpheme denotes an action (V), the following morpheme represents the dominated object (O).

c. *Modification*

In such a combination, the preceding morpheme modifies or restricts the following one.

d. *Subject-Predicate*

The preceding morpheme is a chief part; while the following morpheme gives an account of its status.

e. *Predicate-Complement*[9]

The preceding morpheme denotes a movement or an action, the following morpheme gives the result or tendency for such a movement or action.

f. *Reduplication*

Different from the phone-reduplicate word in the simple word category, the reduplication compound word consists of two morphemes.

(See Examples A.1.8 – A.1.13.)

(ii) Adjunctive form

In this form, the roots denote the basic meaning of the word, and the prefixes as well as suffixes give the additional meaning of the word.

(See Examples A.1.14 – A.1.15.)

[9] The complement that is the adjunctive constituent of a predicate is used to modify the predicate.

Morphological rules play an important role in constructing the internal structure of words. In *classical* Chinese, most morphemes are also words. In *modern* Chinese, however, most words are disyllabic or polysyllabic, and most original free monosyllabic morphemes (can act as a word alone) now change to bound morphemes (which need to be combined) in compound words. Therefore, the study of modern Chinese morphological procedure is necessary for us to automatically segment Chinese words in sentences. After the morphological processes are described, we will deal with another topic in Chinese, namely Chinese grammar, in the next section.

3.3.2 Grammar

In this section, Chinese grammar will be described regarding part-of-speech, different phrases and sentence patterns etc. In particular, the relationships between constituents of sentences are emphasized.

Some characteristics of Chinese grammar make it different from western language grammar (Qian et al., 1995), such as:

- *The sentence construction method is identical to the word formation method*

  Most of the complex forms of the phrases are constructed using the same patterns as compound words, such as, coordination, verb-object, modification, subject-predicate, and predicate-complement. Because the basic forms of the Chinese sentence are composed of different phrases (include extended phrases) and tones (sometimes add particles), the construction of sentences is the same as it is for the word, e.g., 太阳出来了。 (The sun has risen.) Here, 了 (LIAO[10]) is a mood auxiliary word (the explanation refers to the next page) which is a kind of particle.

- *Nominal kernel sentence and predicate kernel sentence*

  There are two large classes of sentences, that is, *nominal kernel sentence* and *predicate kernel sentence*. The former sentence is based on nouns. The latter class, however, is based on verbs or adjectives. For example, 早晨阳光明媚。 (In the morning, the sun was shining brightly.) is a nominal kernel sentence. 早晨 (morning) is its kernel or head; while 她的手冰凉。 (Her hands are icy cold) is a subject-predicate sentence and belongs to the predicate kernel sentence class. In the Chinese sentence, an adjective can be also used as a predicate. Here, 冰凉 (be icy cold) is a predicate and kernel.

- *The topical position of a Chinese sentence*

  Generally speaking, the topic is *prior to* the subject of sentences, e.g., 这件衣服我洗好了。 (I washed these clothes.) Here, 这件衣服 (these clothes) is the topic.

- *In general, the modifiers are prior to the heads in sentences*

  The attribute is the modifier for a nominal kernel sentence, and the adverbial is the modifier for a predicate kernel sentence. For instance, 那是一个刚参加工作不久的天真幼稚的青年。 (That is a naive and childish youth who

[10] It is difficult to translate a Chinese auxiliary word into English, we use the phonetic spelling to note this here.

has just begun employment.) In this sentence, 刚参加工作不久的 (has just begun employment), and 天真幼稚的 (naive and childish) are two modifiers of the noun 青年 (youth); 昨天下午在图书馆里我刚看见他。(Yesterday afternoon I just saw him in the library.) Here, 昨天下午 (yesterday afternoon), 在图书馆里 (in the library) and 刚 (just) are three adverbials as the modifier for the verb 看见 (saw).

- *A structure class is not strictly identical to a function class*

  In Chinese grammar, a part-of-speech does not strictly correspond to a determined constituent in sentences. For example, a verb can be used as a subject or an object. An adjective can be used not only as an attribute, but also as a predicate, a complement, an adverbial, a subject or an object. Additionally, a noun can also be used as an attribute. Because Chinese words have no flections (morphological change), the word forms used as different constituents in sentences are the same.

- *Some special part-of-speeches*

  In Chinese, auxiliary words play an important role in connecting linguistic units, labeling structural relations, and indicating aspect and mood (Li and Thompson, 1981). The *structure* auxiliary words, such as 的 (DE), 地 (DI), and 得 (DE), indicate the subordinative relationships between sentence constituents. The *aspect* auxiliary words, such as 着 (ZHE), 了, and 过 (GUO), denote the continuous, perfective, and experiential aspect, respectively. The *mood* auxiliary words, such as 的, 了, 吗 (MA), and 呢 (NE), express different moods of sentences. In addition, the Chinese *quantifier* is also a special part-of-speech. The numeral and nominal quantifiers are combined to form quantifier phrases. It is joined with the following noun, e.g., 一台机器 (one machine). Here, 台 (TAI) is a quantifier related with the noun 机器 (machine). The numeral and verbal quantifier are combined to form the movement number, e.g., 北海公园我去过五次。(I have been to Beihai Park for five times.) 五次 (five times) is the movement number of the verb 去 (been to).

- *Some special predicate phrase structures*

  In Chinese, the predicate-complement structure is often used to denote an action and its result. This structure is made up of two verbs or a verb and an adjective. The category combination between verb and complement is very free. For instance, 洗干净 (wash + clean); 洗破了 (wash + ragged + LE); 洗晚了 (wash + late + LE); 把他洗哭了 (BA + him + wash + cry + LE) etc. The verb-object phrase structure is another commonly used predicate structure, which is varied and complex. There exist many collocation constraints between verb and object. Besides the two kinds of phrases mentioned above, the *serial* verb phrase means that two or more verbs following the subject occur in a sentence, depending on their time sequence. Note that there is no marker indicating what the relationship is between them (Li and Thompson, 1981).

Chinese words are divided into two large classes, that is, the *full word* (content word) and the *particle* (function word). The former class cannot be enumerated exhaustively; the latter class, however, contains only a countable number of function words. In Chinese, the full word class consists of nouns, verbs, adjectives and numerals. Both verbs and adjectives can act as a predicate. Additionally, the particle class is mainly composed of quantifiers, localizers, volitive words, directional words, adverbs, pronouns, propositions, conjunctions, auxiliary words, interjections, etc. Note that the full word can be put in any position and is regarded as a main constituent of a sentence. But the particle must be bound to the full word, hence its position in sentences is not free. In general, it can only act as a secondary constituent of a sentence.

Chinese phrase structures can be divided into two types. One is the combination of full words. Another is the combination between full words and particles. Both types can be further categorized into several subtypes that are given as follows:

(i) The combination phrase of full words

a. *Coordination Phrase*

b. *Verb-Object Phrase*

c. *Modification Phrase*

d. *Subject-Predicate Phrase*

e. *Predicate-Complement Phrase*

f. *Apposition Phrase*

(See Examples A.2.1 – A.2.6.)

(ii) The combination phrase between full words and particles

a. *Quantifier Phrase*

b. *Locative Phrase*

c. *Prepositional Phrase*

d. *Auxiliary Phrase*

e. *Volitive Phrase*

f. *Directional Phrase*

(See Examples A.2.7 – A.2.12.)

The expansion of phrases is the combination of two or more phrases into a new complex phrase. There are three types in different combinations:

(i) Expansion of *Coordination*

There are two expansion forms of coordination. One is for nouns, and another is for attributes as the modifiers that occur in front of the modified noun.

Formal expression:

NP → $N_1 + N_2 + ... + N_m$.

NP → $AM_1 + AM_2 + … + AM_m + NM_1 + NM_2 + … + NM_n$ + N;

AM_i → ADJ + DE, i = 1, 2,…, m;

NM_j → N, j = 1, 2,…, n.

Note that AM and NM mean an adjective and a nominal modifier respectively.

Example 3.1

NP → $N_1 + N_2 + N_3$ → 小学 + 中学 + 大学 (primary school(s) + middle school(s) + university(universities)). The representation of the phrase structure is shown in Figure 3.1:

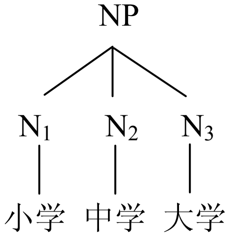


Figure 3.1 Noun Coordination Phrase

Example 3.2

NP → $AM_1 + AM_2 + NM_1$ + N → ADJ + DE + ADJ + DE + N + N → 年轻 + 的 + 男 + 的 + 语文 + 教师 (young + DE + male + DE + Chinese language + teacher). There are three modifiers in this phrase (i.e., *multiple-modifier* phrase). The corresponding phrase structure is described in Figure 3.2.

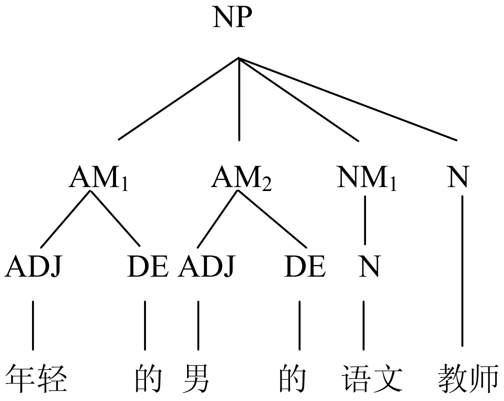


Figure 3.2 Modifier Coordination Phrase

(ii) Expansion of Hierarchy

With the combination of attributes, the hierarchical levels of the phrase structure are extended.

Formal expression:

$NM_2 \rightarrow NM_1 + N_2$;

$NM_3 \rightarrow NM_2 + N_3$;

…

$NM_m \rightarrow NM_{m-1} + N_m + DE$;

$NM_1 \rightarrow N_1$;

$NP \rightarrow NM_m + N$.

Example 3.3

$NM_1 \rightarrow N_1 \rightarrow$ 中国 (Chinese); $NM_2 \rightarrow NM_1 + N_2 \rightarrow$ 中国 + 人民 (Chinese + people); $NM_3 \rightarrow NM_2 + N_3 + DE \rightarrow$ 中国人民 + 智慧 + 的 (Chinese people + wisdom + DE); $NP \rightarrow NM_3 + N \rightarrow$ 中国人民智慧的 + 结晶 (Chinese people's wisdom + crystallization). Its phrase structure is represented in Figure 3.3.

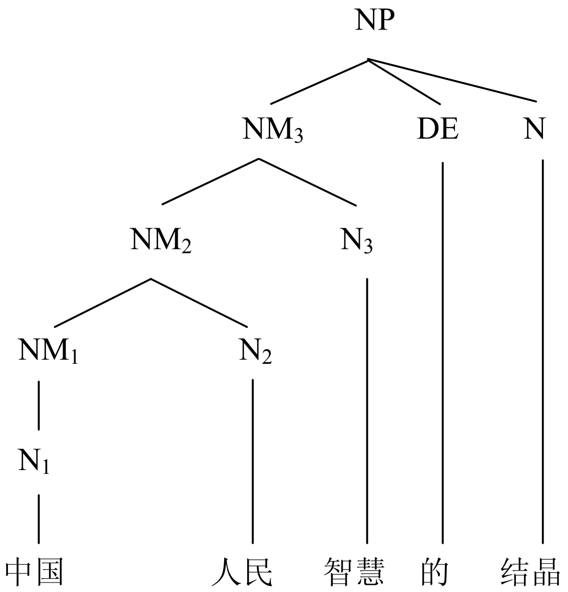


Figure 3.3 Modifier Combination Phrase

(iii) Expansion of Kernel

In general, the order of verbal phrases in the extended phrase indicates the action's sequence.

Formal expression:

$VP_i = V_i + N_i$; $i = 1, 2, \ldots, m+1$

$NP = VP_1 + VP_2 + \ldots + VP_m + V$

$NP = VP_1 + VP_2 + \ldots + VP_m + VP_{m+1}$

Example 3.4

VP_1 =上 (go) + 街 (street); VP_2 = 买 (buy) + 菜 (vegetables); V = 回来 (come back); NP = VP_1 + VP_2 + V = 上街买菜回来 (go to street and buy vegetables, then come back). The phrase structure is specified in Figure 3.4.

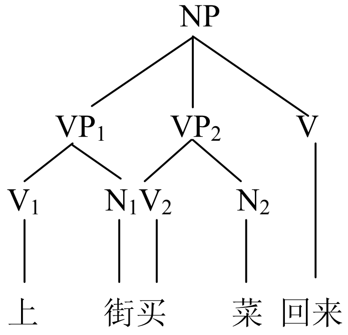


Figure 3.4 Verbal Phrase Coordination

The sentence is one of linguistic units, which is rather important for grammatical analysis (Chao, 1968). The sentence pattern is a structure model by which a speaker organizes grammar units. Based on these models, speakers can construct different sentences from grammar units. At the same time, s/he can also apply the expansion and dislocation rules for deriving more varied and colorful sentences.

When determining to which pattern a sentence belongs, we want to check the kernel of this sentence structure and grammar function of the kernel. Regarding the sentence structures, all sentences can be divided into two classes, that is, the *nominal kernel* group and the *predicate kernel* group. These two classes correspond to nominal kernel based sentence and predicate kernel based sentence respectively. Figure 3.5 is a system of modern Chinese sentence patterns (Qian et al., 1995).

In order to depict the different features of the above patterns, we provide the following descriptions to distinguish their different functions in use.

(i) *Nominal Sentence* (N Sentence)

This sentence is constructed by nouns or the expansion of nouns that act as the kernel of a sentence structure.

(ii) *Adverbial-Nominal Sentence* (AN Sentence)

Generally speaking, adverbs only modify predicates or adjectives, but do not modify nouns. However, as nouns and nominal phrases can also be regarded

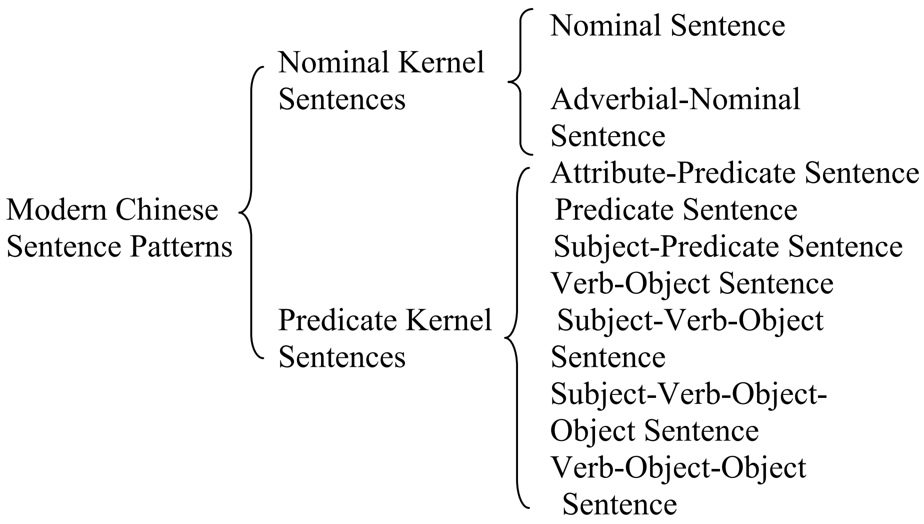

Modern Chinese Sentence Patterns
Nominal Kernel Sentences
Nominal Sentence
Adverbial-Nominal Sentence
Predicate Kernel Sentences
Attribute-Predicate Sentence
Predicate Sentence
Subject-Predicate Sentence
Verb-Object Sentence
Subject-Verb-Object Sentence
Subject-Verb-Object-Object Sentence
Verb-Object-Object Sentence

Figure 3.5 Modern Chinese Sentence Patterns

as an independent clause like predicates, they can also be modified by adverbs in AN sentences.

(iii) *Attribute-Predicate Sentence* (AP Sentence)

In modern Chinese, an attribute can modify not only nouns (This attribute is the expansion constituent of the noun. Viewing the function as a whole, the extended phrase is the same as the noun.), but also predicates (This attribute is not the expansion constituent for the predicate of the head. From the functional point of view, the constructed phrase is not identical to the kernel predicate of the structure while it seems to construct a nominal phrase.) in AP sentences.

(iv) *Predicate Sentence* (P Sentence)

The P sentence is constructed by predicates or the expansion of predicates.

(v) *Subject-Predicate Sentence* (SP Sentence)

This is a kind of sentence that consists only of subject and predicate. Apart from the case of predicative adjective and intransitive verb, a sentence with a VO-form verb as the predicate also does not have an object.

(vi) *Verb-Object Sentence* (VO Sentence)

Sometimes the subject is omitted in a VO sentence, which usually is of practical use in Chinese. In general, the meaning of these sentences does not have a direct relationship with the wording. Some VO sentences are a kind of conventional speech.

(vii) *Subject-Verb-Object Sentence* (SVO Sentence)

This is the most important sentence pattern in modern Chinese. SVO sentences with an action verb as sentence structure kernel are a basic structure of such a pattern. Sometimes a predicate or predicate phrase is an object. In addition, this pattern can be simplified into a SV, VO, V or N sentence. But such a sentence relies to a large extent upon context.

(viii) *Subject-Verb-Object-Object Sentence* (SVOO Sentence)

Some verbs must have two objects. The type (person, thing, etc.) of objects should be dependent on the verb in the sentence. Generally, the object near the verb is called *near-object* (i.e., *indirect object*) that represents person(s). Another object is called *far-object* (i.e., *direct object*) that denotes thing. But some verbs cannot collocate only with near-object or far-object.

(ix) *Verb-Object-Object Sentence* (VOO Sentence)

It doesn't include a subject or omits the subject in certain contexts.

(See Examples A.2.13 – A.2.21.)

As we mentioned above, summarily, Chinese grammar is rather different from western language grammars in part-of-speech, phrase and sentence pattern. Therefore, we use Chinese grammar as a guideline applied to our research work, especially in the processing for part-of-speech tagging, ontology building, named entity and relation recognition.

3.3.3 Semantics

3.3.3.1 Semantic Units and Semantic Field[11]

The earliest practical work of semantic study was annotating ancient books. Philology[12] of Chinese gradually grew out of this work. During that period, Chinese semantic study was called *Xun Gu Study* which has a history of more than 2,000 years. Ancient researchers not only annotated ancient books, but also compiled guide books for annotation. Although Xun Gu Study was only to aid in the research of ancient written language, it is still an important reference for the study of modern Chinese semantics.

Generally speaking, it is known that the language model consists of three levels, that is, the *pronunciation*, the *grammar* and the *semantics*. Each level is an independent system. The Chinese semantic system is composed of many semantic units that exist in *paradigmatical* relationships as well as a number of semantic units that exist in *syntagmatic* relationships. The semantic system is different from the pronunciation and grammar system:

- It belongs to the spirit world and can not be directly observed;
- It is an *open* system and has the property of being both *relatively stable* and *often variable*;
- Among the three systems, the semantic system has the *most* units and the *most complicated* relationships between the units;
- It is closely related to the complicated subject and object world that a language reflects.

The semantic system is constructed on the basis of pronunciation and grammar systems. In order to load and distinguish the semantics, the pronunciation system provides the synthetic

[11] Actually, a semantic field represents a semantic taxonomy.

[12] The semantics of developing periods can be divided into philology, traditional semantics, and modern semantics (Wu, 1999).

methods of the phoneme and the syllable. Similarly, for the sake of organizing and expressing the semantics, the grammar system provides the rules for building words, phrases and sentences. Therefore, the pronunciation and grammar are the carriers of the semantics. The semantics is the content of the communication; while the pronunciation and grammar are the means of it.

There are seven semantic units in Chinese: sememe, semantic component, morpheme semantics, semantic cluster, sentence semantics, discourse production semantics, and implicit semantics (Wu, 1999). These units indicate the meaning or content units of the Chinese language. The sememe consists of the semantic component that is the smallest semantic unit, e.g., the sememe for the word 丈夫 (husband) is (近亲属 close relative) ←→ (配偶关系 spouse relation) + (男性 male). Here, 'close relative', 'spouse relation' and 'male' are semantic components. '←→' indicates that the spouse relation is of equal relation. '+' denotes yes, that is, male; The morpheme semantics means Chinese word meaning; the semantic cluster is the meaning of a fixed or free phrase and is often composed of several sememes. e.g., 抛砖引玉 (throw a sprat to catch a whale) consists of four sememes and is a fixed phrase (i.e., it is an idiom.); The sentence semantics means the meaning at a sentence level; the discourse production semantics is the *largest* semantic unit and deals with the meaning of a talk, a text or a book. It is the research object of the *suprasentential* semantics; the implicit semantics is an additive meaning for the sememe and the sentence semantics. The relationships between the semantic units, the pronunciation and the grammar can be shown in the following figure:

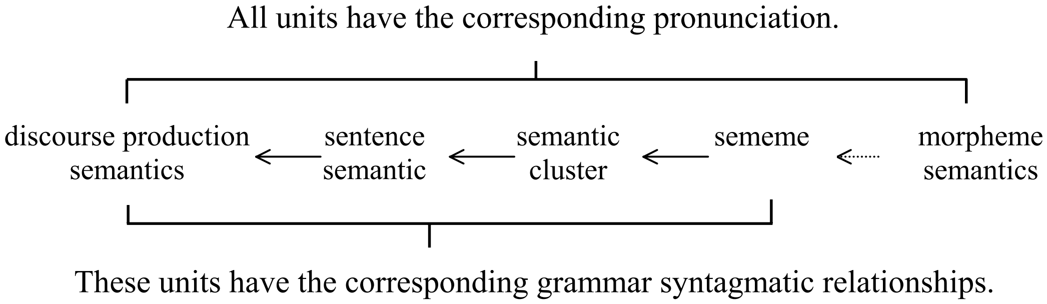

All units have the corresponding pronunciation.
discourse production semantics
sentence semantic
semantic cluster
sememe
morpheme semantics
These units have the corresponding grammar syntagmatic relationships.

Figure 3.6 Semantic Units, Pronunciation, and Grammar

In fact, each semantic unit does not exactly correspond to a pronunciational unit. For instance, a Chinese syllable may correspond not only to the morpheme semantics, but also to the sememe. The speech flow includes pronunciation and semantics. The above five semantic units that are separated from speech flow are attached to respective pronunciations. On the other hand, a sentence is composed of several phrases that consist of several words of speech depending on the grammar. Similarly, sentence semantics is composed of several semantic clusters which consist of sememes. Note that the sememe does not consist of morpheme semantics (see the dotted arrowhead). The morpheme semantics only prompts the sememe or constructs implicit semantics. The discourse production semantics can be a sentence semantics, or the meaning of a sentence group or some sentence groups.

The analysis for semantic components is a paradigmatical analysis. In the analysis, we can find the paradigmatical relationship which reflects how to define a sememe using semantic components through the comparison of different sememes.

Example 3.5 Analysis of the sememes “鞋” (shoes), “靴子” (boots) and “袜子” (socks).

(i) *Determination of the analyzed semantic field*:

A semantic field is a hierarchy of sub-semantic fields, which have a set of complete sememe in semantics. The semantic component analysis starts with the smallest sub-semantic field which can cover all the sememes to be analyzed.

(ii) *Comparison*:

After the determination of the starting point (the smallest sub-semantic field), we may find the related semantic components through the comparison of the sememes. With these semantic components, we can define a sememe by its relationship with the semantic components, that is, the Sememe Structure Expression (SSE) (Leech, 1987; Jia. 1999). The following table is the analyzed result for three sememes, namely, 鞋 (shoes), 靴子 (boots) and 袜子 (socks). In the table, “鞋”, “靴子”, and “袜子” denote three sememes, and 穿在脚上的东西 (object on feet), 走路时着地 (touch ground during walking) and 有筒 (tube-shaped) represent three semantic components. It is equal to the following SSE:

“shoes”: sh(object on feet) t–(tube-shaped) t+(touch ground during walking)

“boots”: sh(object on feet) t+(tube-shaped) t+(touch ground during walking)

“socks”: sh(object on feet) t±(tube-shaped) t–(touch ground during walking)

In the above SSE expressions, ‘sh’ and ‘t’ denote the essence and a specific property respectively. ‘+’ or ‘–’ indicates whether a semantic component has a specific property (yes or no).

	“鞋” (shoes)	“靴子” (boots)	“袜子” (socks)
穿在脚上的东西 (object on feet)	X	X	X
走路时着地 (touch ground during walking)	X	X	
有筒 (tube-shaped)		X	X

Table 3.1 Semantic Component Analysis

Example 3.6 The analysis of the sememes 赢 (win), 和 (draw) and 输 (lose) uses the Modern Chinese Dictionary (The Language Institute of Chinese Academy of Social Sciences, 1999)

Another analysis procedure is to analyze the semantics of Chinese words and to find the suitable semantic components for the sememes. In the Modern Chinese Dictionary, the semantics of these three sememes are described as follows:

赢：在下棋、赛球、赌博等较量中胜利。

Win: achieve victory in a game of chess, a ball game or gambling.

和：在下棋、赛球的较量中不分胜负。

Draw: conclude without either side winning in a game of chess or ball game.

输：在下棋、赛球、赌博等较量中失败。

Lose: being defeated in a game of chess, a ball game or gambling.

From the explanation we can select the appropriate semantic components for SSE, such as 胜利 (victory), 在…中 (in …), 下棋 (play chess), 赛球 (play ball game), 赌博 (gamble), 人 (person) and 队 (team). The SSE is shown as follows (Jia, 1999):

"win"：<d> x{→(victory)f[(in …)(play chess)(play ball game)(gamble) …]} zh[(person)∨(team)]

"draw"：<d> x{→←(victory)f[(in …)(play chess)(play ball game)]} zh[(person)∨(team)]

"lose"：<d> x{←(victory)f[(in …)(play chess)(play ball game)(gamble) …]} zh[(person)∨(team)]

where '<d>' indicates that the sememe represents an action or a behavior; 'zh' represents subjects; 'x' means the subject's behavior, action, movement, or variation; 'f' indicates the scope of application; '→←' denotes the neutralization of two opposite tendencies; '∨' represents disjunction.

The general semantic field of modern Chinese includes a great number of sub-semantic fields. Because of the differences between sememe properties and relationships of sememes, the sub-semantic fields can be divided into several types as follows (The examples give the smallest sub-semantic fields.):

(i) *Classification*: e.g., [traffic_tools (车 (vehicle), 船 (shipping), 飞机 (airplane) and etc.)]

(ii) *Part*: e.g., [constituent_of_tooth (齿根 (root), 齿冠 (top), 齿颈 (neck))]

(iii) *Sequence*: e.g., [examination_marks (优 (excellent), 良 (good), 及格 (pass), 不及格 (fail))]

(iv) *Relationship*: e.g., [teach(教师 (teacher), 学生 (student))]

(v) *Antonymous*: e.g., [human_beings (男人 (man), 女人 (woman))]

(vi) *Polarization*: e.g., [wealth (穷 (poor), 富 (rich))]

(vii) *Partial Negation*: e.g., [degree (绝对 (absolute), 相对 (relative))]

(viii) *Synonymous*: e.g., [begin_to_do (开办 (set up), 兴办 (initiate), 创办 (establish))]

(ix) *Branch*: e.g., [arm_movement (扔 (throw), 抛 (cast), 投 (drop))]

(x) *Description*: e.g., [wet (湿淋淋 (dripping wet), 湿漉漉 (damp))]

Many complicated relationships exist between sub-semantic fields. Those relationships can be classified into vertical and horizontal relationships. The former is the relationship between parent sub-semantic field and child sub-semantic field. The latter is the relationship between child sub-semantic fields under the same parent sub-semantic field. Example A.3.1 in Appendix A shows a semantic architecture of modern Chinese kinship terminologies (Leech, 1987; Jia, 1999).

As shown in Figure A.1, the more generic a sub-semantic field is, the higher its position in the hierarchy. Different from the hierarchical structure shown in this figure, sometimes a sub-semantic field might belong to different upper fields according to different attributes of its sememes. Since Chinese has a very rich vocabulary, this structure can be very complex. Additionally, because of development and change of semantic levels, the general semantic field of the Chinese language is an open system.

3.3.3.2 Sentence Semantics

Sentence semantics is a common semantic unit in a discourse. For example, the simplest discourse production semantics only embodies a kind of sentence semantics which is able to communicate with somebody. According to a certain semantic structure, the sememe and semantic cluster in sentence semantics can be combined. This structure is called Semantic Structure of Sentence (SSS) (Leech, 1987; Jia, 1999). The composition of SSS is described in the following:

(i) *Topic and Comment*

In general, the semantics of a sentence is composed of two elements, an object and an explanation of it. The former is called *topic*, the latter is called *comment*. The topic is known for addressers and addressees. The comment is the facts which an addresser wants to impart an addressee.

(See Example A.3.2.)

(ii) *Constituents of Sentence Semantic Structure*

The typical semantic structure of a sentence consists of four constituents: predicate, argument, description constituent and linking constituent. Of them, the predicate is the most important, and the argument is the most complex.

a. *Predicate*

The predicate is the constituent for describing the topic in sentence semantics. That is, it describes the topic's variation, movement, action or feeling, wish or state, condition, property or its relationship with other objects etc. There exist some few cases in which the predicate is omitted. The predicate is the kernel of a sentence structure. Its type determines the type of sentence structure.

a.1 *Zero Argument*

Such a predicate denotes a kind of variation or movement in nature, and it does not require any argument.

a.2 *One Argument*

It shows the property, status, condition, feeling etc. It needs to combine with an argument in a sentence.

a.3 *Two Arguments*

The predicate requires two arguments to describe the relationship between objects, such as topic and comment.

a.4 *Three Arguments*

It indicates a movement or an action that require three arguments.

(See Examples A.3.3 – A.3.6.)

b. *Argument*

It is a constituent that represents an object, which corresponds to a noun or a pronoun, in sentence semantics. It can be divided into *基本格* (*basic case*) and *一般格* (*general case*).

b.1 *Basic Case*

The basic case can be divided into *主体格* (*nominative*), *客体格* (*accusative*) and *与格* (*dative*).

b.1.1 *Nominative*

A nominative is assigned to the subjects which do something such as movement and action, or have property and state. It can be further divided into *施事格* (*agent*), *主事格* (*possession*), *参与格* (*participation*) and *遭遇格* (*experience*).

b.1.1.1 *Agent*

It is an actor of the predicate (action).

b.1.1.2 *Possession*

It corresponds to the predicate that represents either the relationship between topic and object or property, state and condition of topic.

b.1.1.3 *Participation*

It is one of the participants in an activity or a movement.

b.1.1.4 *Experience*

The predicate represents an action, a variation or a status that is involuntary. In that situation, the subject belongs to the experience case.

(See Examples A.3.7 – A.3.10.)

b.1.2 *Accusative*

An accusative is given to the objects which are the objects of action or movement, or a kind of property or state. It can be subcategorized into *受事格* (*patient*), *结果格* (*result*), *说明格* (*declaration*) and *客事格* (*object*).

b.1.2.1 *Patient*

It is a recipient of the action denoted by a predicate.

b.1.2.2 *Result*

An actor has a desire or a motivation, then she/he does something depending on her/his desire or motive and finally obtains the result.

b.1.2.3 *Declaration*

It describes the related specification of a possession.

b.1.2.4 *Object*

This case indicates an experienced object (person or thing) to correspond to the experience of subjects.

(See Examples A.3.11 – A.3.14.)

b.1.3 *Dative*

It is an indirect object of an action.

(See Example A.3.15.)

b.2 *General Case*

It can be subdivided into *环境格* (*circumstance*), *凭借格* (*means*), *根由格* (*reason*) and *修饰格* (*modification*).

b.2.1 *Circumstance*

It represents some elements in the circumstance of a movement, an action or a status. It can be further divided into *范围格* (*range*), *时间格* (*time*) and *空间格* (*space*).

b.2.1.1 *Range*

It specifies the action range.

b.2.1.2 *Time*

It shows when the action or movement takes place.

b.2.1.3 *Space*

It explains the direction and location of action.

(See Examples A.3.16 – A.3.18.)

b.2.2 *Means*

Some actions are performed with the help of some tools; by using some materials; with a sort of manner; in a certain scale. It can be divided into *工具格* (*tool*), *材料格* (*material*), *方式格* (*manner*) and *基准格* (*scale*).

b.2.2.1 *Tool*

It indicates the tool used by the action.

b.2.2.2 *Material*

It represents the material used by the action.

b.2.2.3 *Manner*

It shows the manner of an action.

b.2.2.4 *Scale*

It indicates the scale of a status or an action.

(See Examples A.3.19 – A.3.22.)

b.2.3 *Reason*

The ground, cause and goal of taking some actions are described by the reason. It can be divided into *依据格* (*basis*), *原因格* (*cause*) and *目的格* (*goal*).

b.2.3.1 *Basis*

It specifies the grounds of the action.

b.2.3.2 *Cause*

It gives the cause of the action.

b.2.3.3 *Goal*

It indicates the goal of the action.

(See Examples A.3.23 – A.3.25.)

b.2.4 *Modification*

The modification is used for modifying arguments. It can be divided into *属格* (*genitive*), *描写格* (*description*) and *同位格* (*apposition*).

b.2.4.1 *Genitive*

It is also called possessive and indicates that an object possesses another object.

b.2.4.2 *Description*

The arguments can be combined in the sentence semantics, in which one modifies another. The modifier belongs to the description.

b.2.4.3 *Apposition*

The apposition (modifier) and the modified argument refer to the same object.

(See Examples A.3.26 – A.3.28.)

c. *Description constituent*

In order to describe the predicate in sentence semantics, we organize a corresponding constituent called *description constituent* to do that. It can be divided into two types:

c.1 *Modifier for Predicate*

There are seven types of such modifiers:

c.1.1 *Mode or Modality*

c.1.2 *Time or Space*

c.1.3 *Degree*

c.1.4 *Range*

c.1.5 *Negation*

c.1.6 *Repeat*

c.1.7 *Mood*

(See Examples A.3.29 – A.3.35.)

c.2 *Complement Constituent of Predicate*

It can be divided into six subtypes:

c.2.1 *Result*

c.2.2 *Time or Space*

c.2.3 *Tool or Material*

c.2.4 *Possibility*

c.2.5 *Degree*

c.2.6 *Quantity of Action or Behavior*

(See Examples A.3.36 – A.3.41.)

d. *Linking constituent*

It joins sememes, semantic clusters, clause semantics, sentence semantics or related constituents of the argument. It represents relationships such as cause, hypothesis, concession, condition, coordination, selection, progression and contrast.

(See Example A.3.42.)

The semantics of a sentence can be depicted in a tree structure. Such structures can be classified as simple, complicated and complex.

(i) *Simple Sentence Semantics*

In general, it corresponds to a logic proposition and can be divided into no predicate, zero-argument predicate, one-argument predicate, two-arguments predicate and three-arguments predicate.

e.g., the sentence in Example A.3.6: 这位售货员递给王小姐一双鞋。The tree structure of this simple sentence semantics is shown in Figure 3.7:

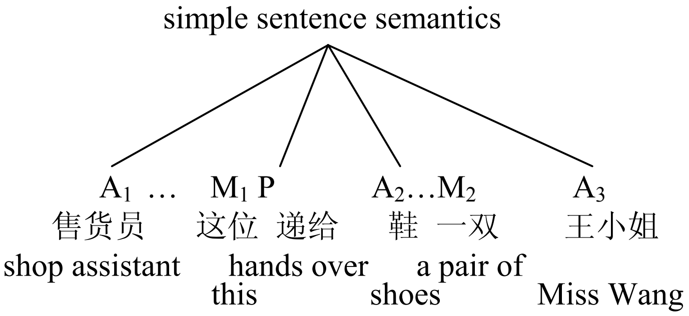


Figure 3.7 Tree Structure of Simple Sentence Semantics

Note: A_1 = A_1(topic, agent); A_2 = A_2(comment, patient); A_3 = A_3(comment, dative); P = P(comment, three arguments); M_1 = M_1 (topic, apposition); M_2 = M_2 (comment, quantity).

(ii) *Complicated Sentence Semantics*

Sometimes some constituents in sentence semantics can also be a sentence semantics. They are called *sentence semantics of constituent*.

e.g., 三班的作业我已经改完了。(I have finished correcting the papers of Class three.) The following figure gives the tree structure of this complicated sentence semantics:

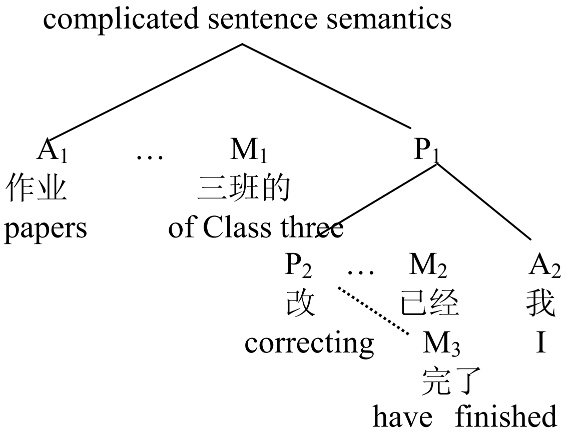


Figure 3.8 Tree Structure of Complicated Sentence Semantics

Note: $A_1 = A_1$(topic, patient); $A_2 = A_2$(comment, agent); $P_1 = P_1$ (comment, simple sentence semantics); $P_2 = P_2$ (comment, one argument); $M_1 = M_1$ (topic, possessive); $M_2 = M_2$ (comment, time); $M_3 = M_3$ (comment, result).

(iii) *Complex Sentence Semantics*

Two or more simple sentence semantics are combined into single sentence semantics by certain semantic relationships called *complex sentence semantics*. In such a complex sentence, the semantics of the simple sentence is called the semantics of sub-sentence.

e.g., 由于天太冷，不少人穿了大衣。(Because it is too cold, many people put on their overcoats.) The tree structure of complex sentence semantics is shown in Figure 3.9:

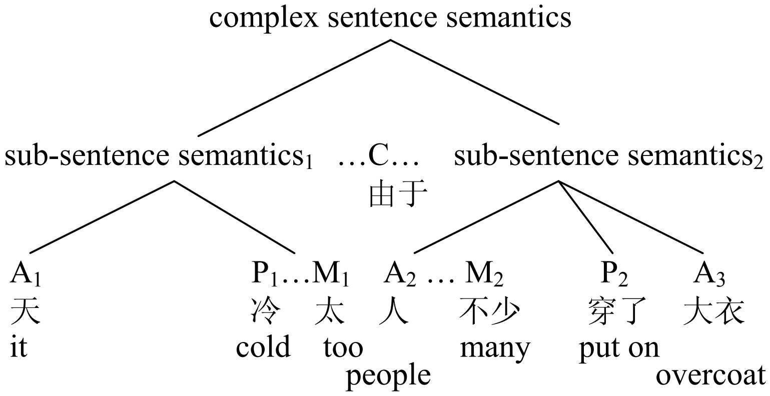


Figure 3.9 Tree Structure of Complex Sentence Semantics

Note: $A_1 = A_1$(topic$_1$, possession); $A_2 = A_2$(topic$_2$, agent); $A_3 = A_3$(comment$_2$, patient); C = Causality; $P_1 = P_1$ (comment$_1$, one argument); $P_2 = P_2$ (comment$_2$, two arguments); $M_1 = M_1$ (comment$_1$, degree); $M_2 = M_2$ (topic$_2$, quantity).

The origin of sentence semantics comes from mental and consciousness. During the expression of sentence semantics, it is attached to a kind of substance forms. For example, Chinese sememes are attached to the pronunciation of content words. Because the pronunciation of content words is already itself a loader for sememe, it can not entirely carry other information simultaneously, such as cases, predicate types, semantic types of the description constituents, and the relationships among sub-sentence semantics in the complex sentence semantics etc. In other words, the pronunciation of content words can not completely load the syntagmatic relationships of sememes in sentence semantics. Therefore, the structure and syntagmatic relationships of Chinese sentence semantics are commonly loaded by particles, word order, tone, meaning of sememes, and context of sentence semantics, which have either practical pronunciation, pronunciation sequence or certain phenomenon. These are all substance forms.

Semantic knowledge is very important for information extraction. Yuan (2002) has put forward the idea that the discourse-text, argument structure, and logic structure knowledge are key knowledge for Chinese information extraction, including:

- discourse structure
- text relationship
- argument structure
- thematic role
- role conversion
- anaphora relation
- negative structure
- scope ambiguity
- operator restriction relation

Although all of the above contents are important for Chinese information extraction, due to limited space we only present knowledge associated with our research work.

3.4 Summary

In this chapter, some related Chinese linguistic knowledge that is the basis of this research work has been presented. As we mentioned above, Chinese is the official language of the Chinese Mainland and Taiwan, and it belongs to the family of Sino-Tibetan languages. In order to reform language, the 20th-century movement in China has resulted in an exciting program of language planning. From 文言文 (literary speech) to 白话文 (colloquial language), 普通话 (common language), and 拼音 (phonetic spelling), these important historical periods have become *milestones* in the development of the Chinese language.

Morphology is related to the internal structure of words. In Chinese, the *monosyllabic morpheme* is the basic form of morphemes and corresponds to a determined character in the written language. Therefore, the overwhelming majority of Chinese characters have a one-to-one correspondence with morphemes. On the other hand, many morphemes might have the same pronunciation, hence in most cases, a Chinese syllabic mode has a one-to-many correspondence with the characters that are homonymous characters. Contrary, multiple tones may be pronounced in some morphemes, namely, polyphonic characters. As regard to word, most of the Chinese words are disyllabic words of two types: simple word and compound word, whose subtypes are shown in Table 3.2.

Simple Word	Compound Word	
unbroken word phone-reduplicated word transliterated word onomatopoetic word	Complex Form	Adjunctive Form
	coordination verb-object modification subject-predicate predicate-complement reduplication	prefix + root root + suffix

Table 3.2 Simple and Compound Words

Morphological procedure plays an important role in constructing the internal structure of words. In classical Chinese, most morphemes are also words. In modern Chinese, however, words have become largely disyllabic or polysyllabic, and formerly free monosyllabic morphemes (each can be as a word alone) now only occur as bound morphemes (need to be combined) in compound words.

Chinese grammar, in this chapter, mainly deals with three topics, namely, part-of-speech, different phrases and sentence patterns.

Generally speaking, Chinese words are divided into two large categories: the *full word* and the *particle*. The former has the *open* property that means new words will be unceasingly added or the word number such as numeral number is unlimited; while the latter is a *closed* set with limited vocabularies. In Chinese, the full word category consists of nouns, verbs, adjectives and numerals. Note that the verb and adjective can all act as a predicate. In addition, the particle category is mainly composed of quantifiers, localizers, volitive words, directional words, adverbs, pronouns, propositions, conjunctions, auxiliary words, interjections etc. It is worth noting that a full word can be located in any position and is regarded as a principal constituent in a sentence. But a particle must be bound to a full word, and its position in a sentence is not free. In general, it only plays the role of a secondary constituent of a sentence.

Regarding Chinese phrase structures, on the whole, we can divide them into two types: one is the combination between full words; another is the combination between full words and particles. Both types can be further separated into several subtypes which are listed in Table 3.3. Additionally, the expansion of phrases, which combines two or more phrases into a new complex phrase, is another combination mode of phrases. In the expansion, there are three kinds of combination mode, that is, *coordination, hierarchy* and *kernel.*

The sentence pattern is a structure model by which a speaker organizes grammar units to construct sentences. When we recognize sentence structure, the structure kernel of a sentence and its grammatical function can help us in judging to which pattern a sentence belongs. All Chinese sentence patterns are enumerated in Table 3.4.

Combination Phrase	
Full Words	Full Words and Particles
Coordination Phrase	Quantifier Phrase
Verb-Object Phrase	Locative Phrase
Modification Phrase	Prepositional Phrase
Subject-Predicate Phrase	Auxiliary Phrase
Predicate-Complement Phrase	Volitive Phrase
Apposition Phrase	Directional Phrase

Table 3.3 Combination Phrase

During the period of Philology, Chinese semantic study was called *Xun Gu Study* which was used to annotate ancient books. Although the Xun Gu Study was about ancient written language, it is still an important reference for studying modern Chinese semantics.

It is known that language model consists of three levels: the *pronunciation*, the *grammar* and the *semantics*. The Chinese semantic system is composed of many semantic units which exist in *paradigmatical* relationships as well as a number of semantic units which exist in *syntagmatic* relationships. The pronunciation and grammar systems serve the semantic system. The pronunciation system gives the synthetic methods of the phoneme and the syllable. The grammar system provides the rules for building words, phrases and sentences.

Modern Chinese Sentence Patterns	
Nominal Kernel Sentences	Predicate Kernel Sentences
Nominal Sentence Adverbial-Nominal Sentence	Attribute-Predicate Sentence Predicate Sentence Subject-Predicate Sentence Verb-Object Sentence Subject-Verb-Object Sentence Subject-Verb-Object-Object Sentence Verb-Object-Object Sentence

Table 3.4 Modern Chinese Sentence Patterns

There are seven semantic units in Chinese language: sememe, semantic component, morpheme semantics, semantic cluster, sentence semantics, discourse production semantics, and implicit semantics. These units indicate the different sizes of meaning blocks of Chinese.

Actually, the analysis for semantic components is a kind of *paradigmatical analysis*. In the analysis, we could find the paradigmatical relationship which reflects how to define a sememe using semantic components through the comparison of different sememes.

The semantic field is an architecture built upon sememes. The general semantic field of modern Chinese includes a great number of sub-semantic fields. Because of the differences for sememe properties and relationships among sememes, the sub-semantic fields can be divided into several types, namely, classification, part, sequence, relationship, antonymous, polarization, partial negation, synonymous, branch, description etc. In general, there are many complicated relationships between semantic fields, which can be divided into vertical relationships and horizontal relationships. The former is the relationship between parent semantic field and child semantic field. The latter is the relationship between child semantic fields which are under the same parent semantic field.

It clearly shows that *sentence semantics* is a key semantic unit in the research of the combination of sememes. Therefore, the *semantic structure of a sentence* is regarded as an absolutely necessary object of investigation. Concretely speaking, the principal semantic structure of a sentence is composed of *topic* and *comment*. Further, both of them consist of some constituents of a sentence semantic structure, which are *predicate*, *argument*, *description constituent* and *linking constituent*. As a constituent specifying the topic of a sentence, the predicate has different types (zero, one, two or three argument(s)), which determines the type of sentence structure. Besides predicate, the argument is another constituent that represents an object which corresponds to a noun or a pronoun in sentence semantics. The cases of an argument can be divided into different types that are elaborately shown in Table 3.5. In addition, the constituent in the Chinese sentence which modifies the predicate is called *description constituent*. Moreover, it has two types: modifier for predicate and complement constituent of predicate. Finally, the constituent called *linking constituent* connects sememes, semantic clusters, clause semantics, sentence semantics or related constituents of the argument. For the sake of clearly analyzing sentence semantics, we can utilize tree structure to depict it, which varies with simple, complicated and complex sentence semantics.

In short, Chinese language knowledge has given us this impression: in some aspects, it is quite different from western languages. For example, the Chinese word has less flection (only adding a suffix to the word root) or a Chinese adjective can act as a predicate etc. Sometimes there exist inconsistent sentences, even though they have the same sentence meaning. Therefore, Chinese word order, particle, tone, meaning of sememes, and context of sentence semantics play very important roles. In our research work, we have noted its differences from western languages and utilized Chinese morphological, grammatical and semantic knowledge to establish our resources and design our algorithms, such as the rules for correcting errors from word segmentation and part-of-speech tagging, Sports Ontology, named entity recognition rules, relation patterns, and machine learning algorithms.

Basic Case	Nominative	Agent Possession Participation Experience	General Case	Circumstance	Range Time Space
	Accusative	Patient Result Declaration Object		Means	Tool Material Manner Scale
	Dative			Reason	Basis Cause Goal
				Modification	Genitive Description Apposition

Table 3.5 Argument Cases

Chapter 4

Error Repair for Chinese Word Segmentation and Part-of-Speech Tagging

4.1 Overview

As already noted in the previous chapter, Chinese is a language without word boundary. Therefore, Chinese word segmentation is a fundamental task for Chinese Natural Language Processing (Chinese NLP) and is also a special problem in Chinese NLP. Chinese text is different from an alphabetic text, e.g., an English text, in that it is composed of character strings from a large character set. Chinese character encoding and input method start the era of electronic Chinese processing. Word segmentation aims to promote this advancement to the word level. Only with words being segmented, can other NLP tasks such as *part-of-speech* (*POS*) *tagging*, *syntactic analysis*, *semantic analysis* be made possible. The general procedure of Chinese text processing is shown in Figure 4.1. For example, the knowledge for syntactic analysis, generally speaking, consists of lexicons and rule bases. A lexicon contains lexical, syntactic and semantic knowledge for words while a syntactic rule is constructed by the knowledge of POS of words, etc. Without the process of word segmentation for Chinese text, the lexicon cannot be built automatically. Furthermore, the syntactic structure cannot be analyzed. In short, word segmentation is a *special* process and one of the *difficult* problems in Chinese NLP.

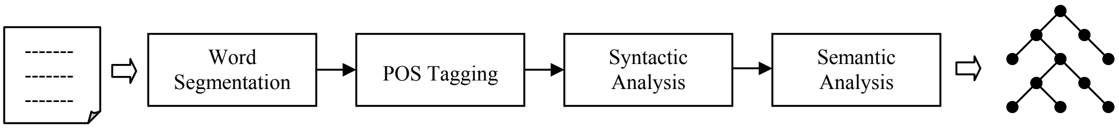


Figure 4.1 Chinese Text Processing Procedure

There are still a number of problems to be solved in Chinese word segmentation. One is the identification for the *boundary* of Chinese words, such as, the word boundary identification between single-character word and morpheme as well as between word and phrase.

Example 4.1: 汉语 (Han Language or Chinese) or 汉 (Han or Chinese) |[13] 语 (Language); 信息抽取(Information Extraction) or 信息 (Information) | 抽取 (Extraction)

[13] Here “ | ” is used as a segmenting symbol.

Some phrases, like that shown below, can be segmented in different ways. Furthermore, the principle for segmentation depends on the domain or application. For instance, there are different word segmentation criteria in different application systems (Liu, 2000):

Example 4.2: 塑胶制品业 (Plasthetics Industry): for *text correcting system*

塑胶 (Plastic) | 制品业 (Products Industry): for *speech recognition system*

塑胶制品 (Plasthetics) | 业 (Industry): for *syntactic analysis system*

In order to maintain a unified and consistent method for Chinese word segmentation, (The Ministry of Mechanics and Electronics Industry of China, 1992) has been used as a national standard for automatic word segmentation. But there still exist some problems, such as indistinct word segmentation specifications, unclear terminologies, word segmentation inconsistency in the examples, and contradiction with linguistics, etc. Thus, sometimes we cannot find a word segmentation criterion that suits our practical usage. To compensate for the shortage of the national standard, we have to work out some word segmentation strategies in accordance with our application domain.

Similarly, after word segmentation, *POS tagging* is another fundamental task for Chinese NLP. There are also some difficulties in Chinese POS tagging. Another peculiarity of Chinese is that characters and words have *no* inflection. Hence the POS of a word cannot be identified with the help of word inflection. Besides, many commonly used words have *multi-POS*, namely, those words can be tagged as different POS in different contexts. Because this kind of word often occurs in texts, there is considerable and complicated disambiguation work to do. Additionally, the human's own judgement makes the situation even more confusing. Below is an example of multi-POS words:

Example 4.3: 讨论 (discuss or discussion) | 是 (be) | 为了(in order to) | 修订 (revise or revision) ｜教育 (educate or education) | 改革 (reform) | 计划 (plan)

In this example, the words "讨论", "修订", "教育", "改革", and "计划" are all multi-POS word. They can be tagged as either *noun* or *verb* depending on their contexts.

Since word segmentation and POS tagging are the basis of Chinese NLP, obviously, we have to consider how to improve their performance (recall and precision) for further NLP tasks. For that there may be two approaches to achieve the above goal:

- Develop a *novel* general Chinese word segmentation and POS tagging system which gives a better performance than current systems or
- *Utilize* a baseline system of good quality and *adapt* it to one application domain.

We have chosen the second approach in our investigation. First of all, the quality of word segmentation and POS tagging is the bottleneck of the quality of Chinese information extraction systems, we cannot dodge this problem. The research of word segmentation and

POS tagging, however, is only a secondary task for us in the project. Secondly, it is *more effective* to improve the quality for word segmentation and POS tagging in a *specific* domain.

In Section 4.2, a Chinese word segmentation and POS tagging system, which we utilize as our baseline system, will be introduced. Then Section 4.3 to 4.5 aim to present an effective error repair approach using a transformation-based error-driven machine learning technique. After that, in Section 4.6, we describe the related experimental condition and results and give the evaluation concerning the error repair approach. Finally, we will summarize this approach and draw a conclusion with regard to its advantages.

4.2 A Chinese Word Segmentation and POS Tagging System

In order to ensure the quality of word segmentation and POS tagging, we compared different existing Chinese word segmentation and POS tagging systems and chose *Modern Chinese Word Segmentation and POS Tagging System* (Liu, 2000) as our baseline system. In this system, the word segmentation component is chiefly divided into five modules as follows:

(i) *Preprocessing*

Identify and markup special symbols from Chinese characters in the text, such symbols can be punctuations, digits, alphabets, etc. The original text is segmented into character strings by those symbols.

(ii) *Cutting character sub-string*

In the first run of text scanning, segment the character strings into substrings with the help of a *feature word* library and some *association rules of word*. Feature words mean affixes, auxiliary words, overlapping words (phone-reduplicate words), unbroken words and so on. The association rule is defined by word formation, formant, and collocation etc.

(iii) *Word segmentation*

The goal of the second run of scanning is to segment Chinese character substrings into words depending on the full word library. The full word has definite lexical meaning. An improved MM (Maximum Matching Method) is applied to word segmentation procedure through backtracking and inferential reasoning. If there exists ambiguous combinational structure or rejected segmentation situation, the following two modules, disambiguation and user interaction, can be used alternatively.

(iv) *Disambiguation processing*

According to the prompt information from the last module, use either corresponding disambiguation rules to process ambiguous combinational structures or the general rule to segment type words (e.g., digit).

(v) *User Interaction*

The task includes the maintenance of the content word library, the temporary word library, the feature word library and the rule library, as well as automatic selection of word by inferential reasoning mechanism.

Note that the modules (i) to (iv) run under online conditions; while the module (v) executes an offline interactive processing.

The component of word segmentation in this system is principally based on the AB (Association-Backtracking) algorithm, which relies not only on word libraries, but also makes use of language knowledge as much as possible. It adopts practical word segmentation rules to solve ambiguous structure problems for improving the efficiency and the precision of word segmentation. Therefore, this algorithm is not simple word matching, but the combination of association (used for empty word formation) and backtracking (used for disambiguity).

In addition, the component of POS tagging utilizes the *probability statistic model* as well as *CLAWS*, *VOLSUNGA*, and the corresponding *transmutation algorithms* to tag different POSs. These POS tags include 25 categories, 85 subcategories and 64 punctuations. The following is a brief introduction to the CLAWS and VOLSUNGA algorithm (DeRose, 1988; Liu, 2000).

The algorithm of CLAWS is described as follows:

1) Establish a single-POS word lexicon and a multi-POS word lexicon. Based on these two lexicons, build a matrix of *collocation probabilities* that indicates the relative likelihood of co-occurrence of all ordered pairs of POS tags.

2) *Select spans and the optimal path*

   Tag candidates of POS for each word in the text according to the two lexicons mentioned above. A span is a word sequence that is composed of n adjacent multi-POS words and their preceding and following single-POS words. The number of multi-POS words in a span is defined as the length of the span. Each such assignment of POS tags in a span is called a *path*. The collocation probability of every pair of adjacent POS tags in a path can be obtained from the collocation matrix. One may thus approximate each path's probability by the product of the probabilities of all its collocations. Each path corresponds to a unique assignment of POS to all words within a span. The paths constitute a span network, and the path of maximal probability is called an *optimal* path that contains the most suitable POS.

The major differences between the VOLSUNGA and CLAWS algorithm are outlined below:

- The optimal path is defined to be the one whose component collocations multiple out to the highest probability. The more complex definition applied by CLAWS, using the sum of all paths at each node of the network, is not used.
- VOLSUNGA overcomes the Non-Polynomial complexity of CLAWS. Because of this change, it is not necessary to resort to a fallback algorithm, and the program is far smaller. Furthermore, testing the algorithm on extensive texts is not prohibitively expensive.
- VOLSUNGA implements *Relative Tag Probabilities* (RTPs) in a more quantitative manner, based upon counts from the Brown Corpus, where CLAWS scales probabilities by 1/2 for RTP < 0.1, and by 1/8 for $p < 0.01$. It uses the RTP value itself as a factor in the equation which defines probability.

- VOLSUNGA uses no tag triples and no idioms. Because of this, manually constructing special case lists is not necessary.
- The version of VOLSUNGA at that time was designed for use with a complete dictionary. Thus, unknown words are handled in a simplistic way.

This system utilizes both linguistic and statistical knowledge for word segmentation and POS tagging. On the basis of that, three new components, the recognition for Chinese personal name, Chinese location name and foreign name, have been added to the system (Liu, 2001).

4.3 Types of Errors

When using the above system to process news in the sports domain, we found there exist *considerable* word segmentation and POS tagging errors. Most of the errors fall under the following categories (Yao et al., 2002a):

4.3.1 Word segmentation errors

- A common noun[14] is segmented into a number of character strings or is not separated from other words,
- A verb, together with an auxiliary or an adverb, is grouped in one segment,
- An abbreviated word is combined with other words into one segment,
- An enclitics is segmented together with a numeral,
- An idiom is segmented into different character strings,
- A personal name or a location name is segmented into a number of character strings,
- A date or time word is combined with a conjunction word, or
- A team name, a competition title or a personal identity word is segmented into different character strings etc.

4.3.2 POS tagging errors

- A common noun is tagged as a personal name, a verb or a location name,
- An adverb is tagged as an adjective or a quantifier,
- A numeral is tagged as a general noun or an adverb,
- A pronoun is tagged as a general noun,
- A verb is tagged as a general noun,
- An auxiliary word is tagged as a verb,

[14] A common noun means it is not a proper noun.

- A personal name is tagged as a location name,
- An abbreviated location name is tagged as a general noun, a verb, a proposition, or a personal name,
- A date or time word is tagged as a quantifier,
- A team name is tagged as a personal name, or
- A competition title or a personal identity word is tagged as a verb etc.

We give the following examples to depict some of the types listed above:

Example 4.4: 国|N|足|N|抵|V|沪|N

The word 国足 (national football team) is an abbreviated team name that should not be segmented; while the word 沪 (Hu) is an abbreviated location name for Shanghai, China.

Example 4.5: 美|A|时|N|杯|N|著名|A|企业|N|足球|N|赛|V|开幕|N

美时杯 (Meishi Cup) is a name of "challenge cup" that should not be segmented. 赛 (match) here is a keyword for the competition title whose POS should be a common noun. In contrast, the word 开幕 (open) is not a general noun, but a verb.

Example 4.6: 江津|N4|和|P|李明都|N4|披挂上阵|I

The personal name should be 李明 (Ming Li). The word 都 (all) is an adverb. They should not be put together. In addition, the POS tag of 和 (and) should not be a proposition, but a conjunction.

Example 4.7: 特别|D|提|V|到了|D|年轻|A

提到了 (have mentioned) should not be segmented into 提 (carry; lift; raise etc.) and 到了 (in the end; finally; eventually etc.), but it should be segmented into 提到 (mention) and 了 (auxiliary word). The word 提到 is a verb, the word 了 is an auxiliary word.

Example 4.8: 他|R|在场|V|上前|V|冲|N|后|D|堵|V|的|U

The phrase 在场上前冲后堵 (attack in front and guard behind on the ground) should be segmented into 在 (on), 场上 (ground) and 前冲后堵 (attack in front and guard behind). The first word is a proposition, the second word is a general noun and the third word is an idiom.

Example 4.9: ‘|W|上三路|N5|’|W|进攻|N|经常|D|受阻|V|；|W

The word 上三路 (upper three paths) is not a Chinese location name, but it is a general noun and means an attack strategy of football. Here the word 进攻 (attack) is a verb.

Note that in the above examples, A, C, D, G, I, J, N, N4, N5, N7, R, U, V and W represent an adjective, a conjunction, an adverb, a morpheme, an idiom, an abbreviation, a general noun, a Chinese personal name, a Chinese location name, a transliterated personal or location name, a pronoun, an auxiliary word, a verb and a punctuation respectively.

Apparently, the correctness of word segmentation and POS tagging for named entities and other constituents in sentences has been *adversely* affected by such errors. If this problem is not solved, we can predict that a good performance of information extraction may not be achieved.

4.4 Repair Rules

Generally speaking, the errors in word segmentation and POS tagging are related to the resources and the algorithms of the baseline system, e.g., linguistic and domain knowledge, training corpus's topic distribution (especially in an application domain), the process of ambiguous combinational structures (disambiguation algorithm) etc. Since the errors have various origins, to comprehensively enhance the performance, the adopted approach should be able to discover the positions of errors, determine the types of errors, and produce effective error repair rules.

The *transformation based error-driven* machine learning approach (Brill, 1995) has been employed to repair word segmentation and POS tagging errors in our research, because it is suitable to fix Chinese word segmentation and POS tagging errors as well as to produce effective repairing rules automatically. Moreover, it can modify the POS tagging result according to the user requirements as well.

As to repair rules, we divide the error repair operations of word segmentation into three types, that is, *concat*, *split* and *slide* (Palmer, 1997; Hockenmaier and Brew, 1998). The operation *concat* means the action of combining some characters (or words) that have been wrongly separated; the operation *split* means the action of splitting some words and the operation *slide* means the action of moving some segment positions forward or backward. The rules referring to three types of error repairing operations for word segmentation are defined as follows:

rectify_segmentation_error (*concat*, old_word1 | old_tag1 | old_word2 | old_tag2 | …, concat_number, new_tag, preceding_word | preceding_tag, following_word | following_tag)

rectify_segmentation_error (*split*, old_word | old_tag, split_position1 | split_position2 | …, new_tag1 | new_tag2 | …, preceding_word | preceding_tag, following_word | following_tag) rectify_segmentation_error (*slide*, old_word1 | old_tag1 | old_word2 | old_tag2 | …, slide_direction_length1 | slide_direction_length2 | …, new_tag1 | new_tag2 | …, preceding_word | preceding_tag, following_word | following_tag)

In the above error repair rules, the *preceding_word* and *preceding_tag* as well as the *following_word* and *following_tag* are the context of an error segment (If the rule is applied as a context-free rule, it can only omit the context descriptions.). The *old_word* or *old_tag* denotes the word or tag before error repair; while the *new_word* or *new_tag* represents the word or tag after error repair. The *concat_number* is the number of *concat* operation. In a string of characters, the *split_position* is a position between two characters which is encoded as the number of characters located on the left side of it. For instance, if a *split_position* is equal to *i*, that means the split position lies between *i*-th and *i+1*-th character. The *slide_direction* is either left or right. The *slide* length is calculated by counting the number of characters between the new splitting position and the original one.

We also define the error repair rule for POS tagging:

rectify_tag_error (old_word | old_tag, new_tag, preceding_word | preceding_tag, following_word | following_tag)

Some examples for both kinds of the error repair rule are shown in Appendix B.

The following examples, referring to examples in Section 4.3, explain the usage of error repair rules. The sentences with word segmentation and POS tagging errors are repaired by corresponding error repair rules.

Example 4.10: 国|N|足|N|抵|V|沪|N

rectify_segmentation_error (concat, 国|N|足|N, 1, J, _|_, 抵|V)

rectify_tag_error (沪|N, J, 抵|V, _|_)

国足|J|抵|V|沪|J

Example 4.11: 美|A|时|N|杯|N|著名|A|企业|N|足球|N|赛|V|开幕|N

rectify_segmentation_error (concat, 美|A|时|N|杯|N, 2, N, _|_, 著名|A)

rectify_tag_error (赛|V, N, 足球|N, 开幕|N)

rectify_tag_error (开幕|N, V, 赛|V, _|_)

美时杯|N|著名|A|企业|N|足球|N|赛|N|开幕|V

Example 4.12: 江津|N4|和|P|李明都|N4|披挂上阵|I

rectify_segmentation_error (split, 李明都|N4, 2, N4|D, 和|P,
披挂上阵|I)

rectify_tag_error (和|P, C, 江津|N4, 李明都|N4)

江津|N4|和|C|李明|N4|都|D|披挂上阵|I

Example 4.13: 特别|D|提|V|到了|D|年轻|A

rectify_segmentation_error (slide, 提|V|到了|D, right1, V|U, 特别|D,
年轻|A)

特别|D|提到|V|了|U|年轻|A

Example 4.14: 他|R|在场|V|上前|V|冲|N|后|D|堵|V|的|U

rectify_segmentation_error (slide, 在场|V|上前|V|冲|N|后|D|堵|V,
left1|left1|right2|right1, P|N|I, 他|R, 的
|U)

他|R|在|P|场上|N|前冲后堵|I|的|U

Example 4.15: ‘|W|上三路|N5|’|W|进攻|N|经常|D|受阻|V|；|W

rectify_tag_error (上三路|N5, N, ‘|W, ’|W)

rectify_tag_error (进攻|N, V, ’|W, 经常|D)

‘|W|上三路|N|’|W|进攻|V|经常|D|受阻|V|；|W

In the above rules, we define three *convenient* error repair operations that are suitable for different error types. Additionally, we add *context constraints* in the rules to ensure the accuracy in fixing errors.

The rules with POS constraints are only applicable when the context satisfies those constraints. The example below illustrates the impact of context constraints.

Example 4.16: As the first example, the inputted sentence is shown as follows:

上半场第 8 分钟中远队就发起了进攻。(At the 8th minute in the first half-time, Zhongyuan Team immediately launched an attack.).

The baseline system outputs:

上|U|半场|N|第|K|8|M|分钟|Q|中|F|远|A|队|N|就|D|发起|N|了|U|进攻|N|。|W|.

In this sentence, 中远 (Zhongyuan) is a team name which should not be separated. We can use the *concat* rule "rectify_segmentation_error (*concat*, 中|F|远|A, 1, N, 分钟|Q, 队|N)" to join two characters. In the rule, F and Q represent a directional word and a quantifier, respectively.

To compare word segmentation influenced by different contexts, the inputted sentence of the second example is given below:

他在带球突破中远射破门。(He shot a ball at a long distance and scores the goal after[15] he dribbled the ball and broke through.).

The baseline system's output is:

他|R|在|P|带|N|球|N|突破|N|中|F|远|A|射|V|破门|V|。|W|.

In the above sentence, "中远" is two separated characters "中" (during) and "远" (at a long distance) and they should not be put together. Since the POS tag of "突破" (break through) and "射" (shoot) is V, the above rule which combine these two characters in the first example sentence will not be applied to the second example sentence. The error-repaired results of both sentences are shown as follows:

上半场|DT|第|H|8|M|分钟|DT|中远|N|队|N|就|D|发起|V|了|U|进攻|V|。|W|

他|R|在|P|带球|V|突破|V|中|N|远射|V|破门|V|。|W|

Note that DT, H, M, and P mean a date or time, a prefix, a numeral, and a proposition, respectively. The word segmentation and POS tagging errors in both sentences have been repaired. For instance, the POS tag of "分钟" (minute) is rectified as a DT.

4.5 Machine Learning Algorithm

For word segmentation and POS tagging errors, we primarily adopt the approach of transformation-based error-driven machine learning, which has been applied to different

[15] The Chinese word "中" in this sentence should be translated as "during". But for a whole English sentence it is better to translate it as "after".

applications, e.g., part-of-speech tagging (Brill, 1994), prepositional phrase attachment disambiguation (Brill and Resnik, 1994), bracketing text (Brill, 1993), etc.

The concrete training and testing algorithms are defined by pseudo code as follows (Yao et al., 2002b):

Training Algorithm

```
 1) input the corpus, in which automatic word segmentation and POS tagging
    has been completed by the baseline system, into auto_corp()
 2) input manual annotated corpus that is manually corrected based on the
    above tests into manu_corp()
 3) for (corp_word_posi = 1 to corp_word_size)
 4)     if (curr_seg(auto_corp(corp_word_posi)) !=
            curr_seg(manu_corp(corp_word_posi)))
 5)         rec_seg_err_info(auto_corp(corp_word_posi))
 6)         rec_seg_cor_info(manu_corp(corp_word_posi))
 7)         build_tf_rule(seg_err_info(auto_corp(corp_word_posi)),
            seg_cor_info(manu_corp(corp_word_posi)))
 8)     if (curr_pos_tag(auto_corp(corp_word_posi)) !=
            curr_pos_tag(manu_corp(corp_word_posi)))
 9)         rec_pos_err_info(auto_corp(corp_word_posi))
10)         rec_pos_cor_info(manu_corp(corp_word_posi))
11)         build_tf_rule(pos_err_info(auto_corp(corp_word_posi)),
             pos_cor_info(manu_corp(corp_word_posi)))
12) for (curr_rule_idx = 1 to tf_rule_size)
13)     if (exam_rule_red(tf_rule(curr_rule_idx), cand_rule_lib())
14)         rem(tf_rule(curr_rule_idx))
15)     else
16)         add(tf_rule(curr_rule_idx), cand_rule_lib())
17) for (curr_rule_idx = 1 to cand_rule_size)
18)     train_cand_rules(cand_rule_lib(curr_rule_idx), train_corp())
19) for (curr_rule_idx = 1 to cand_rule_size)
20)     if (score(cand_rule_lib(curr_rule_idx)) >= 1)
21)         add(cand_rule_lib(curr_rule_idx), reg_rule_lib())
22) sort_reg_rules(reg_rule_lib())
```

Where some functions are defined for various purposes: rec_seg_err_info(), rec_seg_cor _info(), rec_pos_err_info(), and rec_pos_cor_info are to record word segmentation and POS tagging *environments* (i.e., transformation *condition* and *action*); build_tf_rule() is used to construct word segmentation and POS tagging repair rules; exam_rule_red() examines the redundancy of rules, namely, whether a new transformation rule is the same as one of the transformation rules in the candidate rule library; add() is to add a transformation rule to the candidate rule library; rem() removes a redundant transformation rule; train_cand_rules() tests each rule from the candidate rule library in the training corpus and record its *score* (*repaired error number*); sort_reg_rules() is used for sorting regular rule library in terms of their score.

In the training algorithm, the errors are detected by comparing manually annotated text and automatically processed text. Simultaneously, the error environments are recorded. Based on such information, the transformation rules are generated and the effective error repair rules are selected. These rules can repair at least one error in the training corpus (this is a threshold used for choosing regular rules.). The rules in the regular rule library are ranked according to the errors they correct.

Testing Algorithm

```
input the corpus, in which automatic word segmentation and POS tagging
has been completed by the baseline system, into auto_corp()
for (corp_word_posi = 1 to corp_word_size)
    for (reg_rule_idx = 1 to seg_reg_wposcc_rule_size)
        rep_word_seg_err(auto_corp(corp_word_posi),
        seg_reg_wposcc_rule_lib(reg_rule_idx))
    for (reg_rule_idx = 1 to seg_reg_pposcc_rule_size)
        rep_word_seg_err(auto_corp(corp_word_posi),
        seg_reg_pposcc_rule_lib(reg_rule_idx))
    for (reg_rule_idx = 1 to seg_reg_wcc_rule_size)
        rep_word_seg_err(auto_corp(corp_word_posi),
        seg_reg_wcc_rule_lib(reg_rule_idx))
    for (reg_rule_idx = 1 to pos_tag_reg_wposcc_rule_size)
        rep_pos_tag_err(auto_corp(corp_word_posi),
        pos_tag_ wposcc_reg_rule_lib(reg_rule_idx))
    for (reg_rule_idx = 1 to pos_tag_reg_pposcc_rule_size)
        rep_pos_tag_err(auto_corp(corp_word_posi),
        pos_tag_reg_ pposcc_rule_lib(reg_rule_idx))
    for (reg_rule_idx = 1 to pos_tag_reg_wcc_rule_size)
        rep_pos_tag_err(auto_corp(corp_word_posi),
        pos_tag_reg_wcc_rule_lib(reg_rule_idx))
```

Note that seg_reg_wposcc_rule_lib(), seg_reg_pposcc_rule_lib(), and seg_reg_wcc_rule_lib() are the error repair rule libraries for word segmentation with *whole POS context constraints, preceding POS context constraints,* and *without context constraints*, respectively. The definition of error repair rule library for POS tagging has the same meaning as the definition of one for word segmentation.

In the testing algorithm, the usage of error repair rules with context constraints has *priority* over those without context constraints, and the usage of error repair rules for word segmentation has *priority* over those for POS tagging. Through experimental observation, this processing sequence can ensure that the rules repair *many more* errors. On the other hand, it can prevent *new* errors occurring during the repair of existing errors, since, in the scope of word segmentation error repair, we carry out POS tagging error repair as well. Conversely, if we first do POS tagging error repair, it may paper over the errors from word segmentation. In addition, the rules with context constrains are to correct individual-character error cases; while ones without context constrains rectify common-character error cases. Therefore, for an error position or tag, the former rules are preferentially used, and the error usage of the latter rules can be avoided. Examples 4.17 and 4.18 give the corresponding explanations.

Example 4.17 (rule usage priority between word segmentation and POS tagging error repair):

The output from the baseline system for the first example sentence in Example 4.16 is "上|U|半场|N|第|K|8|M|分钟|Q|中|F|远|A|队|N|就|D|发起|N|了|U|进攻|N|。|W|". Suppose there are two repair rules for word segmentation and POS tagging errors separately in the libraries:

rectify_segmentation_error (*concat*, 中|F|远|A, 1, N, 分钟|Q, 队|N)

rectify_tag_error (中|F, N, 分钟|Q, 远|A)

If we first use the above rectify_tag_error rule, the sentence is rectified as "上|U|半场|N|第|K|8|M|分钟|Q|中|N|远|A|队|N|就|D|发起|N|了|U|进攻|N|。|W|". But the above rectify_segmentation_error rule cannot discover and repair the error "中|F|远|A", because the preceding POS tag in the context for this error has been changed.

Example 4.18 (rule usage priority between context constraints and without context constraints):

The output from the baseline system for the second example sentence in Example 4.16 is "他|R|在|P|带|N|球|N|突破|N|中|F|远|A|射|V|破门|V|。|W|". Suppose there are two repair rules for POS tagging errors in the libraries:

rectify_tag_error (中|F, N, 突破|N, 远|A)

rectify_tag_error (中|F, A)

If we initially use the second rule without context constraints, the sentence is corrected as "他|R|在|P|带|N|球|N|突破|N|中|A|远|A|射|V|破门|V|。|W|". Obviously, this is a newly occurring error during the repair of existing errors.

4.6 Experimental Results and Evaluation

The experiments for the error repairer which we have presented in previous sections aim at comparing the results of word segmentation and POS tagging with or without this component. By means of such comparison, we can evaluate the effect of the implemented error repairer. The architecture of the component for repairing word segmentation and POS tagging errors is illustrated in Figure 4.2. Within the figure, the *dotted line* shows the flow process for the *training texts*; while the *solid line* is the one for the *testing texts*. Note that among the six types of named entities (see Chapter 6), personal name (PN), date or time (DT) and location name (LN) are tagged by the first component (the baseline system) and repaired by the second component (the error repairer). They are immediately recognized after error repair.

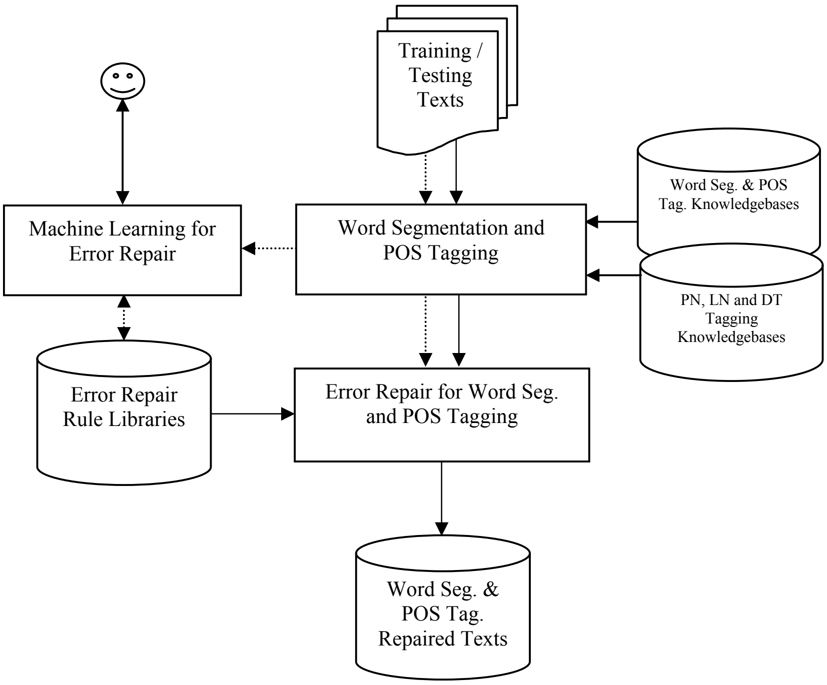

Training / Testing Texts
Machine Learning for Error Repair
Word Segmentation and POS Tagging
Word Seg. & POS Tag. Knowledgebases
PN, LN and DT Tagging Knowledgebases
Error Repair Rule Libraries
Error Repair for Word Seg. and POS Tagging
Word Seg. & POS Tag. Repaired Texts

Figure 4.2 Architecture of the Component for Repairing Word Segmentation and POS Tagging Errors

In the experiments, the training set consists of 94 texts including 3473 sentences (roughly 37077 characters) collected from the football sports news of the *Jie Fang Daily* (http://www.jfdaily.com/) in *2001*. During manual error-correction, we adopted a double-person annotation method, namely, one person is responsible for correction, and another person for correction checking. If there is any difference of opinion for word segmentation and POS tagging, they must consult with or solicit the opinion of other persons, such as asking for opinions from Shan Xi University in China. After learning, we obtain 4304 error repair rules. Among them, 2491 rules are for word segmentation, and 1813 rules are for POS tagging. There are 1730 *concat*, 554 *split*, and 207 *slide* rules in the error repair rule set for word segmentation. Subsequently, we manually distinguish the above rule set into context-sensitive or context-free categories. Consequentially, 790, 315 and 77 are *concat*, *split*, and *slide* context-sensitive rules respectively; while 940, 239 and 130 are *concat*, *split*, and *slide* context-free rules separately. In the error repair rule set for POS tagging, 1052 are context-sensitive rules, and 761 are context-free rules. The testing set is a separate set that contains 20 texts including 658 sentences (roughly 8340 characters). The texts in the testing set have been randomly chosen from the *Jie Fang Daily* from *May 2002*, from which also comes from football sports news.

The evaluation of the performance for the error repairer is composed of four measures, namely, recall, precision, F-measure, and error repair rate. Based on (Palmer, 1997), we define the computing formulas 4.1 to 4.4 for them:

$$\text{Recall} = \frac{\text{correct machine-segmented (tagged) word number}}{\text{hand-segmented (tagged) word number}} \tag{4.1}$$

$$\text{Precision} = \frac{\text{correct machine-segmented (tagged) word number}}{\text{machine-segmented (tagged) character and word number}} \tag{4.2}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

$$\text{Error Repair Rate} = \frac{\text{corrected machine-segmented (tagged) error number}}{\text{total machine-segmented (tagged) error number}} \tag{4.4}$$

Note that in formula 4.2, machine-segmented (tagged) character means that this character cannot independently become a modern Chinese word (see the last paragraph of Section 3.3), that is, this is an error of word segmentation or POS tagging.

Table 4.1, 4.2, and 4.3 show the experimental results for the performance of the error repairer depending on the above formulas, among which the results from two different cases are compared in Table 4.1 and 4.2.

	Average Recall	**Average Precision**	**Average F-measure**
Without Error Correction	91.07	84.67	87.75
With Error Correction	95.08	90.74	92.86

Table 4.1 Performance for Word Segmentation

	Average Recall	Average Precision	Average F-measure
Without Error Correction	80.41	74.73	77.47
With Error Correction	92.39	87.75	90.01

Table 4.2 Performance for POS Tagging

	Average Error Correction Rate
Segmentation	39.99
POS Tagging	56.68

Table 4.3 Average Error Correction Rate for Word Segmentation and POS tagging

The above tables indicate that the average F-measure of word segmentation has increased by 5.11%; while that of POS tagging has even increased by 12.54%. The experimental results have shown that the performance of word segmentation and POS tagging has been *distinctly* enhanced.

Although sometimes it is difficult to directly compare different systems due to the differences in training corpus size, corpus domain, testing corpus size, out-of-vocabulary word number etc., we would like to list a number of reference data regarding Chinese word segmentation and POS tagging as a comparison with our approach:

- In (Palmer, 1997), the author reported his experiment for Chinese word segmentation based on a trainable rule-based algorithm: Character as Word (CAW) (initial F-measure: 40.30%; improved F-measure: 78.10%); Maximum Matching (MM) (64.40%; 84.90%); MM + CAW (82.90%; 87.70%); NMSU segmenter (87.90%; 89.60%).
- Hockenmaier and Brew (1998) proposed the adoption of the following three rules to segment Chinese words, namely, simple bigram rules (BIGR), simple bigram rules by training on the corrected version (BIGR-CORR), and trigram rules (TRIGR). The results are: BIGR (initial F-measure: 42.33%; improved F-measure: 87.10%); BIGR-CORR (42.02%; 87.39%); TRIGR (42.02%; 87.855%).
- In the First International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003), the word segmentation testing results (F-measure[16]) for the CTB (University

[16] The F-measure is calculated based on the closed test that participants could only use training material from the training data for the particular corpus being tested on. No other material is allowed.

Penn. Chinese Treebank) and PK (Beijing University) corpora, which are composed of simplified Chinese codes, are from 73.20% to 88.10% (5 sites) and from 89.40% to 95.10% (8 sites) respectively.

- Zhang et. al. (2003) described the official evaluation for their system ICTCLAS on the sports domain (one of six tested domains): Word Number (33,348); SEG (97.01%); TAG1 (86.77%); RTAG (89.31%). Note: SEG = the number of correctly segmented words / the number of words; TAG1 = the number of correctly tagged 24-tag POS / the number of words; RTAG = TAG / SEG * 100%.

By and large, we may make the judgement that the performance of our approach is roughly comparable to that of the above systems. In other words, like these stand-alone systems, the error repair approach we have adopted is also *appropriate* and *effective*.

4.7 Discussion

In this chapter, we have discussed a *special* problem of Chinese NLP, namely, Chinese word segmentation and POS tagging. Due to the lack of the word boundary in Chinese, we have to segment words in sentences before performing consequent Chinese NLP tasks. Similarly, POS tagging is also a fundamental task following word segmentation. Therefore, the quality of word segmentation and POS tagging is very important for Chinese NLP. In our investigation, in order to reduce both kinds of errors, we decided to adopt an error repair approach to achieve this goal. First, it is necessary to enhance basic quality for the whole quality of the system. Second, it is possible to improve the quality for word segmentation and POS tagging in accordance with a specific domain.

Inspired by the above idea, we proposed an error-repaired approach based on transformation-based error-driven machine learning. It can automatically fix error positions, induce error repair rules, and repair different types of errors.

Some features are defined for learning, such as error and correct word segmentation position, error and correct POS tag, as well as context, etc. It is important that we utilize the context to help us locate error positions for individual-character error cases. In addition, three types of repair rules for word segmentation errors and one type for POS tagging errors have been defined. Using these rules, we can reassign the word segmentation position and replace an error tag with a correct tag.

In the training algorithm, the error positions are determined by comparison of manually annotated text and automatically processed text. Moreover, the error environments are recorded. Based on such information, the candidate transformation rules are generated and the regular error repair rules are selected from them in terms of their *score* (*error repair number*) in the training corpus. In order to use these rules with *priority*, the rules in the regular rule library are sorted. In the testing algorithm, the usage of error repair rules with context constraints are *prior* to those without context constraints, the employment of error repair rules for word segmentation have *priority* over those for POS tagging. By experimental analysis, this performing sequence can ensure that the rules repair *many more* errors. On the other hand, it can prevent *new* errors occurring to a certain extent during the repair of existing errors.

The advantages of our error repair approach for improving quality of word segmentation and POS tagging are (i) It is *more* effective to apply such an approach in a specific domain; (ii) It has *less* development work than if we had developed a new word segmentation and POS tagging system; (iii) Using context-sensitive and context-insensitive error repair rules sequentially, we can take account of both *individual-character* and *general-character* error cases and correct them effectively.

The developing experience and experimental results have proved the above advantages. It is more important that the performance of word segmentation and POS tagging has been *significantly* improved.

Chapter 5

Ontology for Sports Domain

5.1 Overview

Generally speaking, an *ontology* defines a common vocabulary for users who need to share information in a domain. It embodies machine-interpretable definitions of basic concepts in the domain and relations among them (Hoy and McGuinness, 2001). In recent years, not only the development of ontologies has been moving from IT laboratories to the laptops of domain experts, but also ontologies have become popular on the Web such as Yahoo for categorizing Web sites and Amazon for categorizations of books for sale etc. In addition, some tools have been used to support ontology design and development. For instance, the WWW Consortium (W3C) has just implemented the *Web Ontology Language OWL* (Harmelen et al., 2003), a semantic markup language for publishing and sharing ontologies on the World Wide Web. OWL is developed as a vocabulary extension of *RDF* (the Resource Description Framework) (Brickley and Guha, 2000) and is derived from the *DAML+OIL Web Ontology Language* (Connolly et al., 2001); Stanford Medical Informatics (2003) has developed a new version (version 1.9) of the ontology editor and knowledge acquisition system *Protégé-2000* etc.

An ontology benefits the following different areas (Uschold and Gruninger, 1996):

- *Communication*: effective communication within and between people and their organizations.
- *Inter-Operability*: achieve exchangeability for systems by translating between different modeling methods, paradigms, languages and software tools.
- *System Engineering*:
  - Re-Usability: an ontology, i.e., a shared understanding is the basis for a formal encoding of the important entities, attributes, processes and their inter-relationships in a domain. This formal representation may be a re-usable and/or shared component in a software system.
  - Reliability: A formal representation also makes possible the *automation of consistency checking* resulting in more reliable software.
  - Specification: the shared understanding can help the process of identifying *requirements* and defining a *specification* for an IT system. This is especially true when the requirements involve different groups using different terminologies in the same domain, or multiple domains.

Usually, an information extraction system also needs an ontology to provide domain knowledge for identifying different extracted objects such as named entities, named entity relations, and even events, etc. In the *CHINERIS* (*Chinese Named Entity and Relation*

Identification System) system (see Chapter 7), we integrated *Sports Ontology* which is a lexical ontology and is developed under Protégé-2000. Our modeling of Sports Ontology has the following characteristics:

- Utilizing the *re-usability* of an ontology. We integrate a portion of words, phrases and the corresponding concept descriptions from the knowledge dictionary of *HowNet* (Dong and Dong, 2000), a bilingual (Chinese and English) common-sense knowledge base (we can also regard it as an ontology) to Sports Ontology;
- The hierarchical taxonomy mainly consists of three top-concept categories, namely *Object*, *Movement*, and *Property*, which construct respective concept category architectures and correspond separately to *nominal*, *verbal*, and *modifying* words or phrases in the domain lexicon;
- In addition to concept information from HowNet, analogous to FrameNet (Baker et al., 1998), we add *valence* information including *argument cases* (see Section 5.4.2), *named entity categories*[17], and *usage constraints of auxiliary constituents* such as prepositional usage constraints in Movement concept categories;
- In the structure of *Object* concept category, the emphasized relationship is *hyponym* relationship while in the structure of *Movement* concept category, the hyponym relationship can be used to count semantic distance between two concepts. Moreover, the relationships between *Movement* and *Object* concept categories are considered for determining the relationships between predicate and arguments in a sentence; *Property* concept category lays stress on the *modifying* relationships comprising the modification for *Object* and *Movement* concepts.

The development procedure of Sports Ontology is specified in Figure 5.1.

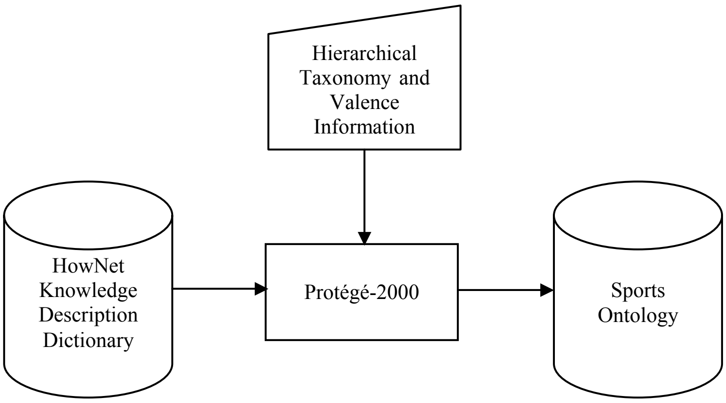

Hierarchical Taxonomy and Valence Information
HowNet Knowledge Description Dictionary
Protégé-2000
Sports Ontology

Figure 5.1 Development Procedure of Sports Ontology

[17] The categories of the named entity defined by us include PN (personal name) , DT (date or time), LN (location name), TN (team name), CT (competition title), and PI (personal identity).

In order to explain what the concept knowledge of HowNet is, we briefly introduce related conceptions of HowNet in Section 5.2. After that, as a whole, Sports Ontology architecture is described in Section 5.3. Based on the above presentation, Section 5.4 gives the definitions of concept descriptions under three top-concept categories. The main points in Section 5.5 specify how to use the information from Sports Ontology in our system. Finally, in Section 5.6, we will discuss the design and application issues of Sports Ontology.

5.2 HowNet

5.2.1 Methodology of HowNet

HowNet is an on-line common-sense knowledge base revealing *inter-conceptual* relations and *inter-attribute* relations of concepts in lexicons of Chinese and their English equivalents. As a knowledge base, the knowledge relationships constructed by HowNet shape a *graph* rather than a *tree* structure. It is devoted to demonstrating the general and specific properties of concepts. For instance, "human being" is the general concept of "doctor" and "patient". The general properties of "human being" are characterized by a series of concept features. Being the agent of cure is the specific attribute of "doctor" while being the experiencer of unwellness is the specific attribute of "patient" and so on. As an example, Figure 5.2 shows a HowNet knowledge graph structure specifying the relationships between concepts such as "doctor", "patient", "disease", "hospital", "medicine", and "expenses".

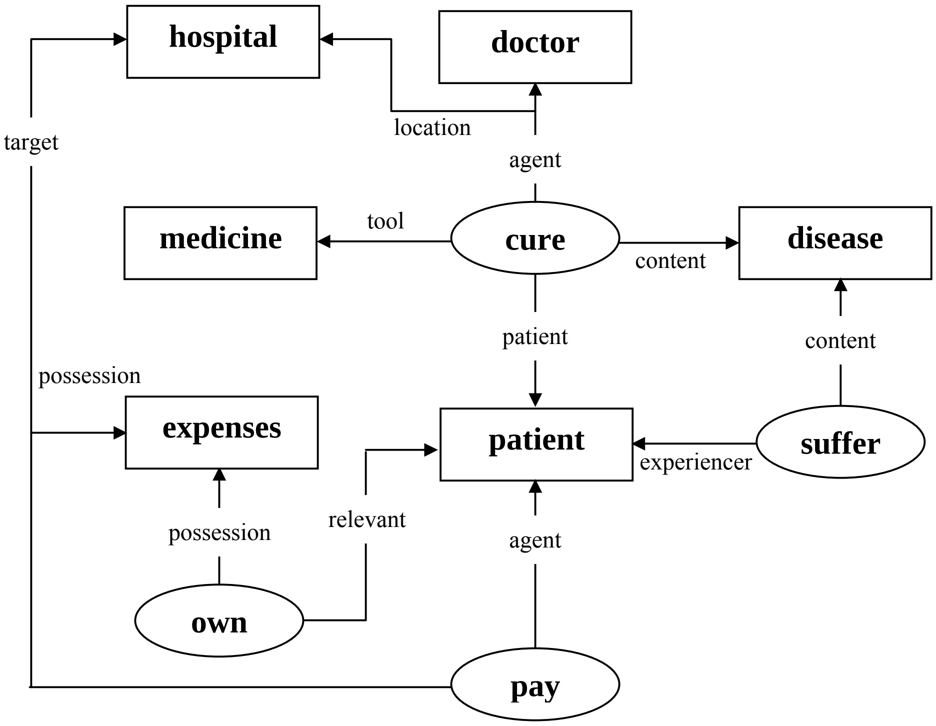


Figure 5.2 HowNet Knowledge Graph Structure

Defining *sememes* is as difficult as defining morpheme (see Chapter 3). The authors of HowNet, however, think that just with morphemes, sememes, though labourious definition, are easily used and understood. Take for instance "human being". Despite being a most complex concept encompassing a set of attributes, it can be regarded as a sememe. They hypothesise that all concepts can be reduced to the relevant sememes and deem further that there exist a closed set of sememes, from which an open set of concepts may be composed. Using the Chinese language to search for this closed set of sememes is really trying a short cut. Approximately, the Chinese characters (including simple words) are a closed set, which can be exploited to express both simple and complex concepts, as well as the inter-concept and inter-attribute connections. The relationships between Chinese words and characters and sememes are illustrated in Figure 5.3.

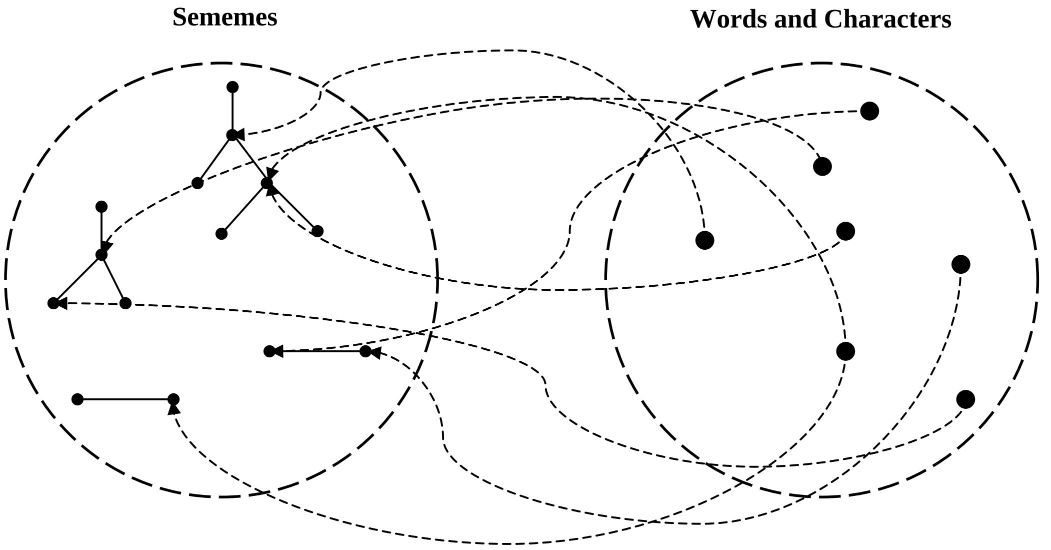


Figure 5.3 Relationships between Chinese Words

The set of sememe is established on meticulous examination of about 6000 Chinese characters. For example, in order to acquire the Event class, the authors even extracted as many as 3200 sememes from Chinese characters (simple morpheme). After the necessary mergence, 1700 sememes were derived for further classification that finally resulted in about 700 sememes. Note that up till this point, *no* Chinese polysyllabic words are involved. These 700-odd sememes then served as a tagging set to tag polysyllabic words, and in the process the authors made *necessary* adjustment and extension when the set cannot satisfy the requirements. Finally the process arrived at a set of over 800 sememes which are now used in HowNet.

Summarily, the building of HowNet is a *bottom-up* grouping approach. The first step is to form a tagging set of sememe through detailed study of all fundamental sememes, and then apply this set to perfect the sememe list. In the next subsection, we want to elaborate a *concept description language* used in HowNet.

5.2.2 Knowledge Dictionary Mark-up Language

First of all, we present two main definitions of HowNet: The one is "*concept*", which is a description of lexical semantics; The other is "*feature*", which is the smallest meaning unit for describing a concept. The word sense in HowNet can be expressed by the combination of several features; Unlike general semantic dictionaries such as *WordNet* (Miller et al., 1990), HowNet does *not* simply map all concepts to a tree-like hierarchical architecture, but tries to use a series of *features* for describing every concept. A detailed structural comparison between HowNet and WordNet can be found in (Wong and Fung, 2002).

The description of concepts in HowNet is an attempt to represent the inter-relation between concepts and that between their attributes. As such, the description is necessarily complex and *unless* a clear set of rules is installed, *consistency* cannot be guaranteed. The description of concepts includes both *general* and *particular* aspects. At the same time, the method of description and the rules concerned must ensure that the inter-concept relations and inter-attributes relations are expressed thoroughly. In this connection, establishing HowNet also implies the design and construction of such a mark-up language. To date, the *Knowledge Dictionary Mark-up Language* (*KDML*) comprises the following three parts:

a) *approximately 1500 features and event roles*;

b) *description symbols*;

c) *word order*.

All the 1500 features are marked bilingually, to order to avoid ambiguity and ensure their readability. They are divided into ten categories:

a) *event*

b) *entity*

c) *attribute*

d) *aValue*

e) *quantity*

f) *qValue*

g) *SecondaryFeature*

h) *syntax*

i) *EventRole*

j) *EventFeatures*

These feature categories can be further divided into three groups: the first one is composed of the categories from a) to g) called "*basic feature*" which is used to describe semantic features of a single concept and is further composed of primary and secondary features; the second one has only a category h) called "*grammar feature*" that depicts grammar features of a word, such as part-of-speech; the third one consists of the categories i) and j) called "*relation feature*", which delineates the relationships between concepts.

Besides features, some symbols applied to describe the semantics of concepts in KDML are listed in Table 5.1.

Symbol	Explanation
,	and
#	related to
%	part of
$	patient, object, possessions, or content of a verbal concept
*	agent or tool of a verbal concept
+	covert role
&	pointer to
~	possible or perhaps have
@	space or time of a verbal concept
?	material of a nominal concept
{}	within which: (i) required role for a verbal concept; or (ii) dynamic role such as prepositional concept
()	within which: a concept word
^	impossible, don't have or cannot
!	a sensitive attribute
[]	between them: common attributes of a concept

Table 5.1 Description Symbols Employed in KDML

In Table 5.1, ",", "~", and "^" express logic relationships between semantic descriptions; while "#", "%", "$", "*", "+", "&", "@", "?", and "!" depict the relationships between concepts. In addition to those, "{}", "()", and "[]" are special description symbols. In the way of example, the concept descriptions of Chinese words from HowNet are shown in Table 5.2.

In the examples of this table we can observe that a concept is defined as a combination of one or more primary features and a sequence of secondary features with prefixed symbols. The primary features indicate the word or phrase's concept category in the HowNet hierarchical taxonomy. Based on concept categories, the secondary features make the word or phrase's concept more *explicit*, even though they are *non-taxonomic*.

The knowledge description outlines in HowNet are highlighted as follows:

- There are two kinds of Chinese words collected by HowNet: one is *full word*; the other is *empty word*;
- The concept for empty words is described by {grammar feature(s)} or {relation feature(s)};
- The concept description for full words, namely semantic description expression, is separated by several commas. There are three types of semantic description expressions:
  a) *Stand-alone* feature description expression: that is, "basic feature", or "concrete word";
  b) *Relation* feature description expression: i.e., "relation feature=basic feature", "relation feature=(concrete word)", or "(relation feature= concrete word)";

Word	Concept Description
比赛（competition)	fact|事情,compete|比赛
教练（coach)	human|人,#occupation|职位,*teach|教
上海（Shanghai)	place|地方,city|市,ProperName|专,(China|中国)
生日（birthday)	time|时间,day|日,@ComeToWorld|问世,$congratulate|祝贺
必须（must)	{modality|语气}
串（bunch)	NounUnit|名量,&(grape|葡萄),&(key|钥匙)
直达（nonstop)	aValue|属性值,property|特性,^CeaseSelfMove|终止自移
布（cloth)	material|材料,?clothing|衣物
诊室（consulting room)	part|部件,%InstitutePlace|场所,+diagnose|诊察,+cure|医治,medical|医
打对折（offer a 50% discount)	subtract|削减,patient=price|价格,commercial|商,(range=50%)
儿童基金会 (UNICEF[18])	part|部件,%institution|机构,politics|政,#young|幼,#fund|资金,(institution|机构=UN|联合国)

Table 5.2 Some Examples of Concept Description from HowNet

c) *Symbol* feature description expression: namely, “relation symbol[19] basic feature” or “relation symbol(concrete word)”;

- In the description of full words, the first expression usually is a primary feature which also is the *most* important expression. In other words, it describes the most basic semantic feature for this full word.

On the whole, there are eight relationships between concepts as well as concept and attribute defined in HowNet:

a) Hypernym-Hyponym, e.g., fruit|水果 → plant|植物;

b) Synonym, e.g., HideTruth|瞒 → MakeNoKnowledge|使不知;

c) Antonym, e.g., clear|清 → blurred|浑;

d) Converse, e.g., appear|出现 → disappear|消失;

e) Part-Whole, e.g., room|房间 → %house|房屋;

f) Attribute-Host, e.g., ampere|安培 → &electricity|电;

g) Material-Product, e.g., tree|树 → ?material|材料;

[18] The United Nations Children's Fund

[19] The relation symbols are used to represent the relationships between concepts and contain “#”, “%”, “$”, “*”, “+”, “&”, “@”, “?”, and “!”.

h) Event-Role, e.g., LandVehicle|车 → *VehicleGo|驶;

In Subsection 5.2.3, *Knowledge Description Dictionary*, a bilingual lexical knowledgebase, is introduced. Every word and phrase concept in this dictionary are described by KDML.

5.2.3 Knowledge Description Dictionary

HowNet Knowledge Description Dictionary is the kernel of the whole system. In this Dictionary, every word or phrase and its concept description etc., form one entry. Regardless of the language types, an entry consists of eight items. Every item is specified by an attribute name and its corresponding value(s) combined by the "=" sign. The left hand side of the "=" sign is the *attribute name*; while the right hand side of that sign is the *attribute value(s)*. The items are arranged in the following sequence:

NO. = word or phrase number

W_C = word or phrase form of Chinese
G_C = word or phrase syntactic class of Chinese
E_C = example of usage for Chinese

W_E = word or phrase form of English
G_E = word or phrase syntactic class of English
E_E = example of usage for English
DEF = concept definition

Below, we would like to provide examples for those words which have more than one meaning. The emphasis is given to *disambiguation* rather than *explanation*. In distinguishing two of meanings of the Chinese word "打" (buy; knit; etc.) for example, one meaning is: "buy|买", and the other is: "weave|辫编". They are found in the Knowledge Description Dictionary as Example 5.1 and 5.2 show:

Example 5.1:

NO.=017174
W_C=打
G_C=V
E_C=~酱油，~张票，~饭，去~瓶酒，醋~来了
W_E=buy
G_E=V
E_E=
DEF=buy|买

Example 5.2:

NO.=017208
W_C=打
G_C=V
E_C=~毛衣，~毛裤，~双毛袜子，~草鞋，~一条围巾，~麻绳，~条辫子
W_E=knit

G_E=V
E_E=
DEF=weave|辫编

Note that E_C indicates several examples of usage for Chinese. For example, "~酱油(~ soy sauce)" expresses "打酱油 (buy soy sauce)". "打毛衣" means to knit a woollen sweater. Besides that, DEF denotes the definition of concept and shall include at least one feature. There is no limitation to the number of features in any DEF, only that the definition is reasonable in content and acceptable in terms of formats. The first item in the DEF shall be a main feature, however, in the case of functional words such as prepositions, conjunctions, sentential adverbs etc., a secondary feature can be used for the first item, but it should be enclosed within {}.

5.3 Sports Ontology Architecture

In recent years, some researchers have utilized HowNet for their investigations and applications: Zhou and Feng (2000) employed three data tables, that is, concept, feature, and relation tables, to build the bi-directions and multi-angles connections among them, as well as integrated all the information in HowNet into a relational network. Thus, for further investigation of Chinese information retrieval and knowledge reasoning, this approach provides an effective means of knowledge acquisition for HowNet. In (Liu and Li, 2002), the authors proposed an approach to computing the word similarity based on HowNet. Moreover, Lai et al. (2002) have constructed a DAML+OIL-compliant *Traditional* Chinese Lexical Ontology in terms of HowNet's taxonomy of primary features and Chinese lexical entries.

Similarly, during the development of Sports Ontology, we also utilized the bilingual lexicon and the corresponding concept information of HowNet. It is designed in terms of the requirements for the identification of named entities and their relations. Sports Ontology is a *lexical* ontology and currently only includes *football* sports knowledge limited to our application domain. In the following we give the definition of Sports Ontology:

Sports Ontology = (Hierarchical Taxonomy, Lexicon, Concept Set, Concept Relation Set)

Where Hierarchical Taxonomy is further defined as follows:

Hierarchical Taxonomy = (Object(Material Object(…), Spirit Object(…), Abstract Object(…), Time(…), Space(…)), Movement(Artificiality(…), Non-Artificiality(…)), Property(Appearance(…), Presentation(…), Color(…), Flavor(…), Nature(…), Ability and Political Integrity(…), Circumstances(…)))

The hierarchical taxonomy of concept categories is illustrated in Figure 5.4. Note that in the figure we omit the names of the lower-level concept categories under three top-concept categories Object, Movement, and Property.

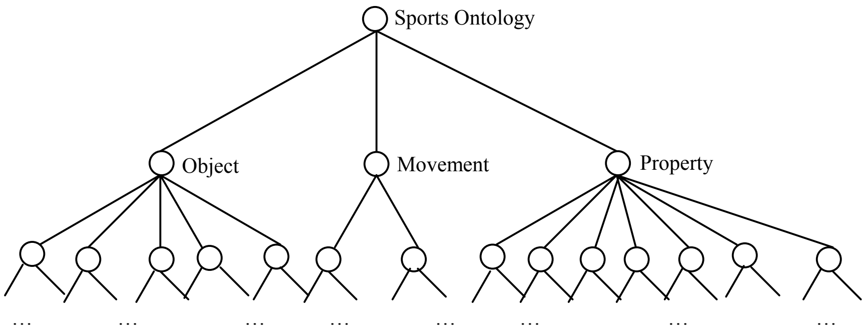


Figure 5.4 Hierarchical Taxonomy of Concept Categories in Sports Ontology

The detailed hierarchical taxonomy of concept categories is also shown in Appendix C[20]. Lexicon is composed of bilingual word entries, i.e., Lexicon = (Chinese Words and Phrases, Corresponding English Words and Phrases). Basically, Concept Set is composed of HowNet lexical concepts. In addition, Concept Relation Set comprises *Object-Object*, *Movement-Object*, *Property-Object*, *Property-Movement*, *Property-Property* relationships. They are distributed in taxonomic categories under three main top-concept categories. For example, in *Movement* concept categories, *argument cases*, *named entity categories*, and *usage constraints of auxiliary constituents* are used to describe Movement-Object concept relation directly or indirectly.

Below, some motivations in designing Sports Ontology are described:

- Words and phrases are linguistic forms of concepts; while concepts are *ideological contents* of words and phrases (Wu, 1999). This is our fundamental philosophy in building such a lexical Sports Ontology;
- In the sports domain, there are different events dealing with *persons*, *organizations*, *equipments*, *competition* (modes, laws, tactics, techniques, honors etc.), *time*, *space*, *actions*, *states*, *properties* and so on (Li, 1995; Yu, 1998). Abstractly, we can divide them into Object, Movement, and Property top-concept categories. According to such a concept classification, a hierarchical taxonomy architecture is formed;
- We combine taxonomic concept relations with *non-taxonomic* concept relations such as *Movement-Object*, *Property-Object*, *Property-Movement* in some concept categories. Thus, it is more convenient and practical to access knowledge in Sports Ontology;
- Obviously, concept relationships depicted by grammar features and relation features are implicit in HowNet. In order to easily process them, we have decided to define explicit concept relationships in concept categories. For instance, a set of argument cases and named entity signs used in the valence of *Movement* concept categories directly describe *Movement-Object* concept relationships.

[20] In the hierarchical taxonomy of Sports Ontology, we have implemented all of the *Movement* concept categories and a part of the *Object* and *Property* concept categories.

In order to further elaborate *Object*, *Movement*, and *Property* concept categories and their applications, in the next two sections we give the definitions of three concept categories and explain how to use this information for recognizing named entities and their relations.

5.4 Concept Descriptions in Sports Ontology

5.4.1 Object Concept Description

The *Object* concepts reflect the concept of material, spirit, time, and space, etc. In general, the concept relationships involved in *Object* concept categories primarily are *Object-Object* taxonomic relations, but sometimes some of the relations go *beyond* the level of concept categories, i.e., it is not sure that the relations must be formed between the children concept categories and the parent concept category. On the other hand, *Object* concepts have predicate-argument relationships with *Movement* concepts and modification relationships with *Property* concepts as well. In the design of Sports Ontology, we describe these two kinds of relationships in *Movement* and *Property* concept categories respectively.

The *Object* concept category is defined as follows:

(Word/Phrase Form; Conceptual Relationships; HowNet-Concept)

Word/Phrase Form is represented by a bilingual word or phrase in the lexicon. Conceptual Relationships mean that an associated word or phrase in Sports Ontology has *trans-level* concept relationships with the word or phrase in Word/Phrase Form. HowNet-Concept is related to the concept of the word or phrase, which is introduced from HowNet. For details, two examples below specify *Relationship* and *Province City* concept categories under *Object* top-concept (see Appendix C.):

Example 5.3:

(观众, spectator; ; human/*look/#entertainment/#sport/*recreation)

Note that this description has no Conceptual Relationships.

Example 5.4:

(杭州, Hangzhou; Zhejiang, China; place/city/ProperName/(China))

In this concept description, we can observe that the *Province City* concept has two main associated concepts: one is the *Province* concept, the other is the *Country* concept. That means Hangzhou (*Province City*) belongs to Zhejiang (*Province*), and Zhejiang is a Province of China (*Country*).

5.4.2 Movement Concept Description

A *Movement* concept expresses the thinking content of verbs. It principally involves *Action*, *Behavior*, *State*, and *Process* (Wu, 1999). In Sports Ontology, for the sake of explicitly depicting the relationships between *Movement* and *Object* concepts, we define argument cases, named entity categories, and HowNet-concepts in concept categories simultaneously. The argument cases we adopt (Lin et al., 1994) indicate semantic types of the nouns or nominal phrases with *Object* concepts dominated by the verbs or verbal phrases having *Movement* concepts in a sentence. Their definitions are listed in Table 5.3. Furthermore, the usage constraints of auxiliary constituents incarnate the relationships between full words and auxiliary constituents, such as the relationships between full words and particles.

The *Movement* concept category is defined in the following:

(Word/Phrase Form; Argument Cases; Usage Constraints of Auxiliary Constituents)

Note that in this concept category Argument Cases and Usage Constraints of Auxiliary Constituents embody the information of HowNet-concepts and named entity categories.

Example 5.4 describes the concept of the word 观看 (watch), which belongs to the *Perception* concept category under *Movement* top-concept category. In the description, *Agent* may be a PN, or a PI such as spectator; *Patient* is a CT. The symbol "-" means that the description of argument case following this symbol is an *optional* item. In other words, it is possible that the *Time* or *Location* case doesn't occur in a sentence at all. In Usage Constraints of Auxiliary Constituents, a particle, e.g., the proposition, 在 (at/in/on) should be used before a word or phrase which is a LN or DT, whose concept is either location or time.

Example 5.4:

(观看, watch; Agent|human/*look/#entertainment/#sport/*recreation|PN, Agent|human/*look/#entertainment/#sport/*recreation|PI, Patient|fact/compete|CT, Time |time|DT; -Location|facilities/@exercise/sport|LN; at/in/on|{location}/{time}| LN/DT; look)

5.4.3 Property Concept Description

The words or phrases regarding *Property* concepts play a role in the modification of the words or phrases with *Object* or *Movement* concepts. According to (Mei et al., 1983), we design *Property* concept architecture. The definition for such a concept category is given as follows:

(Word/Phrase Form; Modification(s); HowNet-Concept)

Modification indicates modified objects, which include the information about top-concept categories, HowNet-concepts, and named entity categories.

Case	Definition
Agent	The subject of a spontaneous movement, action, or status in an event
Essive	The subject of a non-spontaneous movement, action, or status in an event
Possession	The subject that has possessive relationship with an Object in an event
Link	A type, identity, or role of a subject in an event
Patient	An existing direct object related to a spontaneous movement, action in an event
Object	An existing direct object related to a non-spontaneous movement, action in an event
Participation	A constituent of a Possession in an event
Dative	An indirect object which has interests with its subject in an event
Accompaniment	An indirect object that is accompanied or eliminated in an event
Result	Produced, caused, or achieved final result in an event
Scale	An indirect object as a reference to compare or measure a subject in an event
Amount	Involved quantity and frequency in an event
Range	An accompanied state related to a domain or a range in an event
Tool	A used tool in an event
Material	Used material or consumed substance in an event
Manner	Adopted method or form in an event
Basis	The grounds which somebody obeys or counts on in an event
Cause	Reason in an event
Goal	The aim which somebody wants to achieve
Time	The time division when an event takes place or the time interval over which an event takes place
Location	The place or circumstances where an event takes place, or a route from one place to another
Direction	The space-time direction in an event
Degree	The influenced degree (depth and scope) with which an event happens

Table 5.3 Cases Used in Movement Concept

The following example is to describe the word 紧张 (nervous) which belongs to the concept category *Meek* under the concept category *Circumstances* (see Appendix C.). This word can modify words or phrases in *Object* concept categories (*Competition* and *Occupation*) and *Movement* concept category (*Doing*).

Example 5.5:

(紧张, nervous; Object|fact/compete|CT, Object|human/sport|PI, Object|human/sport|PN, Movement|compete/sport; uneasy)

In this section, we give the definitions of concept categories in Sports Ontology and explain how to describe different concepts. For further specifying uses of Sports Ontology, Section 5.5 discusses its application in information extraction.

5.5 Sports Ontology and Information Extraction

Ontologies have possessed many application domains so far, e.g., text annotation (Gan and Wong, 2000), Web agent (Luke et al., 1997), linguistic phenomenon analysis (Bond and Vatikiotis-Bateson, 2002), and bilingual ontology alignment (Ngai et al., 2002) etc. For our Sports Ontology, however, its applications mainly deal with named entity and relation identification.

5.5.1 Identification for Named Entities

There are primarily two applications in named entity identification:

- Determine the *boundary* of named entities;
- Identify the named entities with *special constructions* such as TNs or CTs without the keyword 队 (Team) or 赛 (Competition) separately.

The first application of Sports Ontology is to fix the boundary of named entities e.g., 本年度 (this year) is a phrase and expresses an entire time conception. Therefore, as a whole, it should be identified as a unitary DT. Under the *Nature* concept category there is a word 本 (this) in the concept category *Reference*, its concept description is shown as (本, this; Object|time/year/month/day|DT; aValue/time/now); while the concept description of the word 年度 (year) is expressed as (年度, year; ; time/year). Obviously, the former word is *allowed* to modify the latter word in terms of concept matching, namely, they can be combined. In addition, the other example is the phrase 中国北京 (Beijing, China). It should be identified as a single LN in terms of *Country* and *Special City* concept relationship. The phrase 北京上海 (Beijing and Shanghai), however, should be identified as two LNs depending on the hierarchical concept relationship of *Society*.

The second application of Sports Ontology is to identify TNs or CTs with special constructions. For instance, sometimes there is TNs without keyword in sentences:

Example 5.6: 上视队一球小胜大连。(Shanghai Television Team feebly won Dalian over exactly one ball.)

Example 5.7: 中国取胜瑞典并不容易。(It is not easy that China wants to score a success against Sweden.)

In the sentence of Example 5.6, because 大连 (Dalian) is a city name in China, which is not followed by a keyword, it therefore cannot be identified by the rules of team name recognition (See Chapter 6.). Similarly, in Example 5.7, 中国 (China) and 瑞典 (Sweden) are two country names, neither of which can be identified as two TNs according to the rules.

In Sports Ontology, there are *Action* and *Status* concept categories under *Movement* top-concept category, which include domain verbs such as 胜 (win), 负于 (lose), 逼平 (force to draw), 进攻 (attack), 防守 (guard), 迎战 (take on), 瓦解 (disintegrate) etc., and their corresponding valence constraints. For example, the concrete concept descriptions for the verb 胜 (win) and 取胜 (score a success) are defined as follows:

(胜, win; Agent|human/mass|TN, Patient|human/mass|TN, -Time|time|DT, -Location|place/ProperName|LN, -Degree|aValue/size/small; at/in/on|{location}/{time}|LN/DT; win)

(取胜, score a success; Agent|human/mass|TN, Patient|human/mass|TN, -Time|time|DT, -Location|place/ProperName|LN, -Range|fact/compete; at/in/on|{location}/{time}|LN/DT; win)

According to the above valence constraints, we examine whether the concepts or named entity categories of constituents in both sides of two verbs are *identical*. As seen in Example 5.6, we only find a TN in the sentence, because team names should be *balanced* in the light of the valence constraints of domain verb 胜 (win), therefore we regard the location name 大连 (Dalian) as a candidate of team name. Further, when this candidate is checked, we may know that its constituent is a city name which can be as a constituent of team name. For Example 5.7, when we observe the sentence, both sides of domain verb 取胜 (score a success) only have LNs rather than TNs. Because the domain is *restricted*, the valence constraints require two TNs as predicate verb's agent and patient. Considering a country name can be a constituent making up the team name, we can infer that the true colors of these two LNs are TNs.

5.5.2 Identification for Named Entity Relations

As another aspect of applications, Sports Ontology knowledge can be used for the identification of named entity relations, primarily containing calculation of similarity for the features (see Chapter 7), e.g., valence of verb, concepts of nouns/nominal phrases, adjectives, and adverbs, which are constituents of named entities, and concepts of verbs/verbal phrases. As an explanation, Example 5.8 describes how to compute semantic distance between two verbs by verb concept categories and concept hierarchical structure.

Example 5.8 (Calculate semantic distance between two verbs):

The hierarchical taxonomy of *Movement* concept categories decides different *positions* for each verb in the concept tree structure; the distance between two verbs can be seen as the semantic distance. For instance, the verb 险胜 (win by a narrow margin) and the verb 战平 (draw a battle) lie in concept categories *Win and Lose* and *Draw* separately. Both of the two categories are subconcept categories of the concept category *Relationship*. Therefore, the semantic distance of two verbs is counted as: Win and Lose → Relationship → Draw = 1 + 1 = 2. If we compare another pair of verbs such as 排名 (arrange place in a competition) and 抢点 (tackling), we find that their semantic distance is greater than in the former verb pair. Because they belong to *Status* and *Action* concept categories respectively, the semantic distance is equal to: Existence → Dynamic Status → Status → Artificiality → Action → Motion → Attack = 1 + 1 + 1 + 1 + 1 + 1 = 6. The semantic distance can be used for computing *feature similarity* (see Section 7.4.2).

5.6 Discussion

In this chapter, we have presented the design issues of Sports Ontology:

- According to the *re-usability* principle for ontologies, we utilize information from Knowledge Description Dictionary of HowNet as a basic knowledge source;
- As the surface form of a concept, words have *mutual corresponding* relationships with concepts. This is a starting point for why we design lexical Sports Ontology;
- In addition to the hierarchical taxonomy, we have also constructed the relationships between *Movement* and *Object*, *Movement* and *Property*, as well as *Property* and *Object*. Figure 5.5 illustrates different relationships among them. Note that the arrow direction in the figure indicates the concept relationships from governor to dependent.
- The introduction of argument cases, named entity signs, and usage constraints of auxiliary constituents can *explicitly* build the above *Movement* and *Object* concept relationships. On the other hand, they provide convenient information access condition.

Based on the above design issues, Sports Ontology can basically satisfy the requirement for information extraction. In applications, the information provided primarily is of two kinds: concepts and concept relations. They serve as not only the knowledge to support named entity identification such as the recognition of named entities with special constructions (see Chapter 6) and named entity boundary determination, but also the features to help identify named entity relations (see Chapter 7).

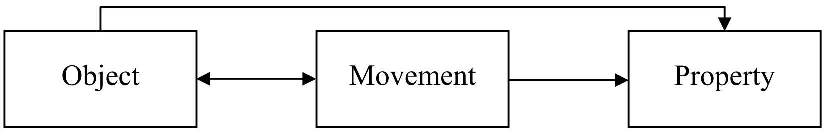


Figure 5.5 Conceptual Relationships in Sports Ontology

Chapter 6

Named Entity Identification

6.1 Overview

As seen in Chapter 2, a series of "Message Understanding Conferences (MUCs)" has remarkably promoted the development of information extraction technology. Among the tasks of MUC-7, *named entity* (*NE*) *identification* is one of the most important tasks of information extraction (Chinchor, 1997). In natural language, a NE represents a person name, a person title, a location name, an organization name, date, time, money, or percentage and so on. Obviously, it is an important syntactic constituent such as subject, object, etc., and semantic constituent, e.g., agent, patient, etc. Therefore, NE identification is also a fundamental task of our investigation.

In the investigation of NE identification, we adopt *football* competition news as our corpus, because a variety of NEs and NE relations exist in the news. Among the NEs, we select six of them as the recognized objects, that is, *personal name* (*PN*), *date or time* (*DT*), *location name* (*LN*), *team name* (*TN*), *competition title* (*CT*) and *personal identity* (*PI*). For example, 莫晨月(Mo Chenyue), 卡洛斯 (Carlos); ９月１９日 (Sept. 19), 本周五 (this Friday), 前７０分钟 (former 70 minutes); 上海(Shanghai), 柏林(Berlin); 中国队 (China Team), 四川队 (Sichuan Team), 上海申花队(Shanghai Shenhua Team), 拜仁慕尼黑队 (Bayern München); 全国女足超级联赛 (National Woman Football Super League Matches), 泰王杯国际足球邀请赛 (Thailand King's Cup International Football Tournament); 门将 (goalkeeper), 前锋 (forward), 外援 (foreign player), 主教练 (chief coach), 裁判员 (judge), 记者 (correspondent), etc.

Although there are different approaches to identifying Chinese NE (Chen et al., 1998; Chen and Chen, 2000; Zhang and Zhou, 2000; Sun et al., 2002), considering Chinese NE's construction as well as the comprehensive superiority in accuracy, efficiency, and robustness of identification, we utilize *Finite-State Cascades* (*FSC*) (Abney, 1996) as a shallow parser to identify different NEs. In order to enhance flexibility and maintainability of this mechanism, we propose an approach to automatically constructing FSC relying upon NE recognition rules. Thus, it is easy to maintain FSC such as addition, modification, deletion of recognition rules. In addition, aiming at special constructions of NEs, we proposed relevant strategies to identify such NEs as well. The architecture of the component for NE identification is shown in Figure 6.1. In this figure, notice that a gazetteer is used for identifying static named entities such as continent, country, province (state), city, club, company, product name, etc., before TN, CT and PI are recognized, because these static named entities are the constituents of TN and CT. Additionally, because FSC provide a multi-level recognition mechanism, we arrange corresponding recognition rule sets in a different NE recognizer. Considering the sequence of NE identification (e.g., in PI recognition rules TN and CT categories are used as POS constraints), we put TN, CT and PI recognizers in low to high order (Another three NEs, namely, PN, DT and LN are immediately recognized after error repair).

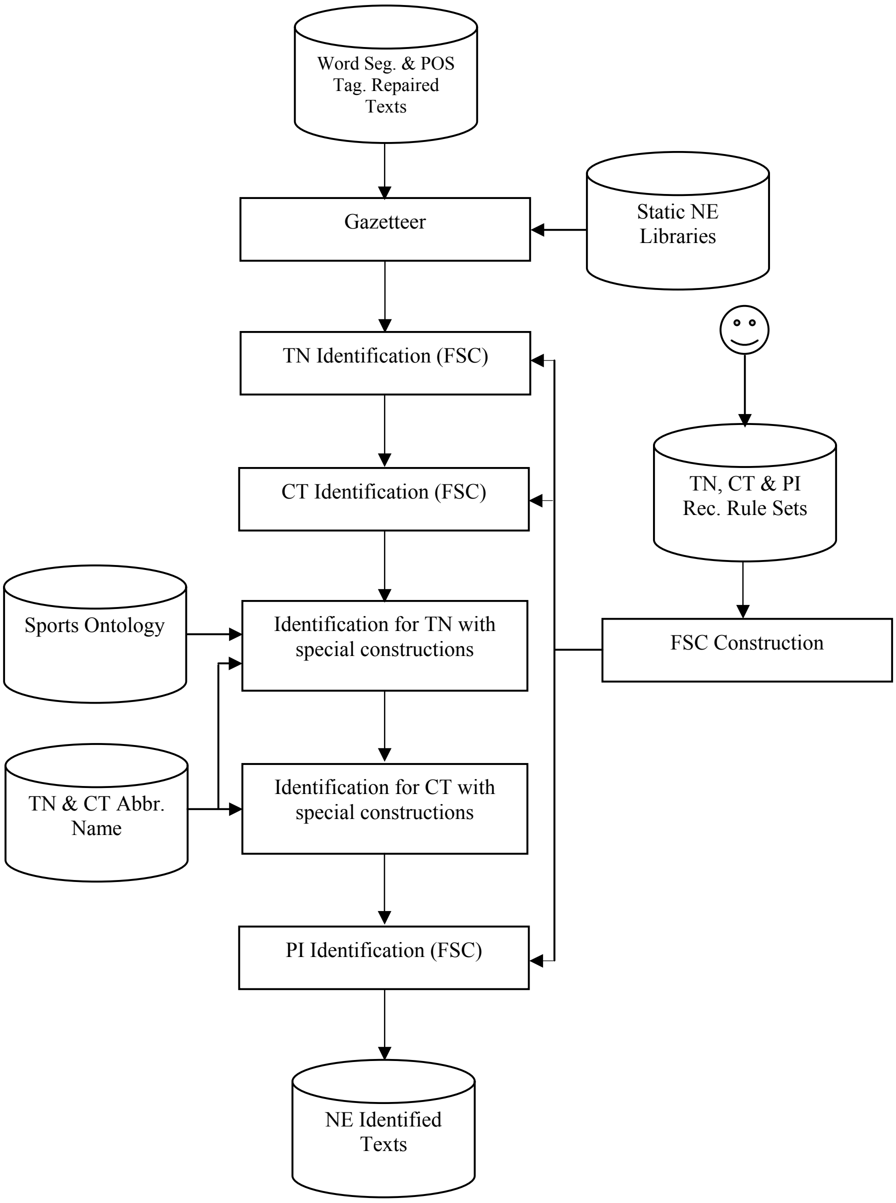


Figure 6.1 Architecture of the Component for NE Identification

The next section introduces the basic conception of FSC, the definition of recognition rules, the formal definitions with regard to FSC, the construction algorithm, and an example concerning automatically constructing a recognizer. In Section 6.3, we describe the procedure of NE identification. Then Section 6.4 puts forward some countermeasures to identify special constructions of NEs. Finally, in Section 6.5, we will discuss the effect of this mechanism in Chinese NE identification and compare our FSC-based parser with that of other systems.

6.2 Automatically Constructed Finite-State Cascades

6.2.1 Finite-State Cascades (FSC)

FSC are a technique used for rapid partial parsing. During the implementation of CASS (Cascades Analysis of Syntactic Structure) System, Abney (1990) applied the embryo of this technique to it. CASS consists of three principal filters, namely the chunk, clause, and parse filters. This parser is a pipeline of simple filters. Each filter makes a definite decision concerning a specific problem, such as part-of-speech (POS) disambiguation or identifying the subject and predicate of simplex clauses. The experiment has proved that the parser is rather fast and reliable. For example, it parsed a megaword from Brown (Kucera and Francis, 1967) and LOB (Johansson et al., 1986) corpora within six hours. On the other hand, it also reaches about 95% accuracy in categorizing and segmenting chunks as well as in labelling chunks as subject, predicate, or neither of these.

In recent years, FSC technique has been applied to a number of information extraction systems, shallow text processing systems, and even development environments for text engineering, e.g., *FASTUS* (*Finite State Automaton Text Understanding System*) (Appelt et al., 1993; Hobbs et al., 1996), *SproUT* (*Shallow Processing with Unification and Typed Feature Structures*) (Becker et al., 2002), *GATE* (*General Architecture for Text Engineering*) (Cunningham et al., 1996), *CCSP* (*Cascaded Chinese Syntactic Parser*) (Zhang, 1998) etc. Among them, the SproUT system can compile regular expressions of Chinese grammar and yield relevant automata and the CCSP system as a shallow parser can analyze different Chinese phrase and sentence structures.

FSC are composed of several levels. Sentence constituents at one level are established on sentence constituents at the previous level, and there is no recursion: constituents never contain same-level or higher-level constituents. In parsing, FSC consist of a series of finite transductions which are represented by T_i. Each T_i is defined by a set of patterns that include a category and a regular expression. The regular expression is translated into a finite-state automaton, all the combined pattern automata produce a single, deterministic, and finite-state level recognizer T_i in which each final state is related to a unique pattern. The recognizer T_i 's input is L_{i-1}, it produces L_i as an output. If the recognizer enters a final state at more than one position in the input, only the longest match creates an output phrase. On the other hand, if the recognizer cannot reach a final state, that is, there is an input constituent which cannot match any patterns in this recognizer, this constituent is simply outputted. Figure 6.2 illustrates the working flow of FSC with n levels. Example 6.1 explains the regular expression and describes parsing procedure.

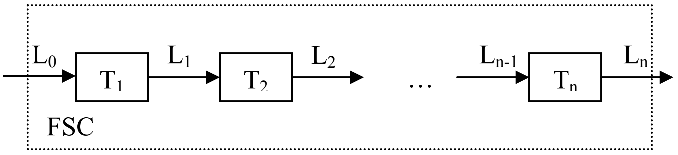


Figure 6.2 Finite-State Cascades with n Levels

Example 6.1:

Suppose there are three transductions in FSC:

$T_1 = \{NP \rightarrow N^* \ N \mid PRON\};$

$$T_2 = \{PP \rightarrow PREP\ NP\}; \qquad (6.1)$$
$$T_3 = \{S \rightarrow PP^*\ NP\ PP^*\ V\ NP\ PP^*\ W\}$$

When the Chinese sentence 他们在运动场看足球比赛。(They watched a football match in a stadium.) is an input, the following figure shows a parsing procedure using the FSC. Note that in the above regular expressions, the syntactic categories S, NP, PP, N, V, PREP, PRON, and W represent sentence, nominal phrase, preposition phrase, noun, verb, preposition, pronoun, and punctuation mark, respectively. Suppose X and Y are two syntactic categories, X* means that X is repeated zero or more than zero times. X Y denotes the sequence of X and Y, namely, X lies in the front of Y. X | Y indicates that either X or Y occurs in the regular expression. The operator priority from high to low is repeat ("*"), sequence, and selection ("|") operator in turn.

In Figure 6.3, the recognizer T_1 begins at word 0 in level L_0. It reaches a final state associated with the NP pattern in position 1, and outputs an NP from 0 to 1 at L_1. The recognizer is then restarted at position 1. No transition is possible, so PREP is simply outputted. Starting from position 2, it reaches a final state related to the NP pattern in position 3, and outputs an NP from 2 to 3 at L_1. As a new starting position, there is no transition in position 3. Thus, V is immediately outputted. Note that in position 4, there are two possibilities for matching the pattern of T_1: the final states associated with the NP pattern are reached at either position 5 or 6. Taking the longest match, T_1 outputs a NP from position 4 to position 6 at L_1. The final parsing step of T_1 is also to simply output W. The recognizer T_2 accepts L_1 and scans it from left to right. It only finds a transition from position 1 to position 3, that is, it reaches a final state related to the PP pattern in position 3, and outputs a PP from 1 to 3 at L_2. Other categories are outputted in the original form at L_2. Finally, the recognizer T_3 reaches a final state related to S pattern and outputs an S from position 0 to position 7 at L_3.

```
L3        ------------------------------------S-----------------------------------
   T3
L2        NP       ---------PP--------    V           NP          W
   T2
L1      --NP --    PREP    --NP--         V       ------NP------      W
   T1
L0      PRON     PREP       N           V       N         N       W

        他们       在      运动场      看      足球      比赛      。
        (they)    (in)   (stadium)  (watch)  (football)  (match)   (.)
    0          1      2          3       4          5        6     7
```

Figure 6.3 Parsing Procedure of FSC

Summarily, FSC have the following advantages:

- *Its multi-level regular expressions are suitable to expressing the level structure of a grammar*

  Generally speaking, syntactic structures can be divided into different levels, such as sentence, clause, phrase, POS, and word level. FSC can conveniently analyze such syntactic structures, because its regular expressions in different levels represent the corresponding grammar level.

- *It can process the center-embedding linguistic phenomenon within finite levels*

  Due to the multi-level structure of FSC, it is possible to simulate the center-embedding in finite levels by utilizing the level relationship and the repetition of regular expressions. For example, in the low level, there exists a regular expression associated with S, whose subject is NP; while in the high level, there is also a regular expression associated with S, but its subject is S of the low level. Thus, FSC realize two-fold center-embedding of S. In fact, the embedding number has to depend on FSC's level number and regular expression organization, that is to say, it only processed finite center-embedding. Because the center-embedding number of practical sentences is usually limited, FSC can be used in real-world applications (Church, 1980).

- *It reduces repetition and redundancy of regular expressions*

  Using the mechanism of multi-level regular expressions, it is easy to do modulation of grammar. A number of common and basic regular expressions are arranged in low-level recognizers; while complex regular expressions are installed in high-level recognizers, so that high-level recognizers can share low-level regular expressions.

- *It can effectively resolve ambiguity*

  Although FSC' parsing mechanism has no feature unification and context constraints like other parsing methods, it provides a very distinct description mechanism for grammar levels. With this mechanism, the strategy of "first easy and difficult afterwards" is applied to parsing. After easy syntactic structures are determined, they amount to rich context information provided for high-level recognizer to parse complex syntactic structure. Therefore, FSC have the ability to resolve ambiguity. In addition, it is easy to add other disambiguous mechanisms in multi-level structure.

In order to reliably and accurately identify NEs, we choose FSC as a shallow parser to do that. However, we are confronted with the construction problem of FSC. Generally speaking, FSC translated by regular expressions are constructed by standard compiler techniques (Abney, 1997; Zhang, 1998) or other techniques, e.g., *FSM* (*Finite State Machine*)*-toolbox-based regular expression compiler* (Piskorski et al., 2002). These expressions, however, merely include a single constraint symbol[21] such as POS constraint. If regular expressions embody complex constraint symbols, e.g., both POS and semantic constraint symbols, it is difficult to adopt the above techniques to construct FSC. In our recognition rules (see next subsection), sometimes a POS constraint symbol may correspond to multiple semantic constraint symbols in a rule. Therefore, for the sake of convenience of construction and maintainability for FSC, we propose an approach to *automatically* construct FSC using NE recognition rule sets. In Subsection 6.2.2, the definition of recognition rules is represented. Then Subsection 6.2.3 discusses the construction algorithm. Finally, in Subsection 6.2.4, an example for building a deterministic finite automaton will be given.

[21] The single constraint symbol means that the symbol in a regular expression is an *atomic* symbol.

6.2.2 Definition of Recognition Rules

In order to construct FSC for NE identification, we define the representation of the NE recognition rules as follows:

$$\text{Recognition Category} \rightarrow \text{POS Constraint} \mid \text{Semantic Constraint}_1 \mid \text{Semantic Constraint}_2 \mid \ldots \mid \text{Semantic Constraint}_n \qquad (6.2)$$

The left-hand side (LHS) of the rule is a recognition category that indicates a recognized NE; while the right-hand side (RHS) of the rule lists POS constraint and its corresponding one or more semantic constraints. Meanwhile, the symbol "|" denotes a separation between constraints. Additionally, the symbol "→" between LHS and RHS represents a productive (or conventional) relationship of these two sides. The rule tag set contains 19 POS tags and 29 semantic tags. A POS constraint describes the part-of-speech and sequence of the named entity's constituents. In addition, a semantic constraint gives the meaning of corresponding part-of-speech or constituents, such as country name, province (state) name, city name, company name, club name, and product name, etc. Example 6.2 specifies a rule for TN identification.

Example 6.2:

TN → N + KEY_WORD | AbbreviationName + TeamNameKeyword | CityName + TeamNameKeyword | CompanyName + TeamNameKeyword | ClubName + TeamNameKeyword | CountryName + TeamNameKeyword | ProductName + TeamNameKeyword | TNOtherName + TeamNameKeyword

Where given X and Y are POS or semantic constraint tags. X + Y denotes the sequence of X and Y, namely, X is ranked in front of Y. This rule means that N (noun) must be an abbreviation name, a city name, a company name, a club name, country name, a product name, or another team name. Following N, a KEY_WORD must occur. Notice that KEY_WORD represents the trigger word within a team name, such as the words 队 (Team), 联队 (League), etc. Of course, sometimes there is no such trigger word in a team name. In this case, we will provide the appropriate identifying strategies (see Section 6.4).

Other examples with respect to the NE recognition rules are listed in Appendix D.

6.2.3 Construction Algorithm

In order to clearly express some fundamental concepts used in our construction algorithm, we give the following definitions. Notice that they extend the definitions given by (Piskorski, 2002).

Definition 6.1: A finite-state automaton (FSA) is a 6-tuple $M = (Q, S_1, S_2, \delta, i, F)$, where Q is a finite set of states; $i \in Q$ is the initial state; $F \subseteq Q$ is a set of final states; S_1 and S_2 are two finite character string sets; and δ: $Q \times S_1 \times S_2 \rightarrow 2^Q$ is the transition function. Moreover, δ is extended to δ': $Q \times S_1^* \times S_2^* \rightarrow 2^Q$ for accepting strings over $S_1 \times S_2$. The language accepted by M is defined as L(M):

$$L(M) = \{(u, v) \in S_1^* \times S_2^* \mid \delta'(i, (u, v)) \cap F \neq \emptyset\} \quad (6.3)$$

An automaton can be represented as a directed graph, in which the nodes indicate states and the edges depict transitions. Notice that we use positive natural numbers for labeling the nodes. Additionally, we use (*u*, *v*) for marking edges, which means *u* and *v* are accepted only when they are simultaneously inputted strings. Finally, we represent the initial and final states as two concentric circles.

Example 6.3: Let M = (Q_M, S_{M1}, S_{M2}, δ_M, i_M, F_M) be an FSA, where Q_M = {0, 1, 2}; S_{M1} = {B, KEY_WORD, J, N}; S_{M2} = {AbbreviationName, CompetitionTitleKeyword, CTOtherName, Range, Rank}; δ_M = {(0, (B, Rank), 1), (0, (J, AbbreviationName), 1), (0, (N, CTOtherName), 1), (0, (N, Range), 1), (1, (KEY_WORD, CompetitionTitleKeyword), 2)}; i_M = 0; and F_M = {2}.

The relevant graph of the automaton M is shown in Figure 6.4. In addition, it accepts the language:

L(M) = {(B, Rank)(KEY_WORD, CompetitionTitleKeyword), (J, Abbreviation Name)(KEY_WORD, CompetitionTitleKeyword), (N, CTOtherName)(KEY_ WORD, CompetitionTitleKeyword), (N, Range)(KEY_WORD, CompetitionTitle Keyword)}.

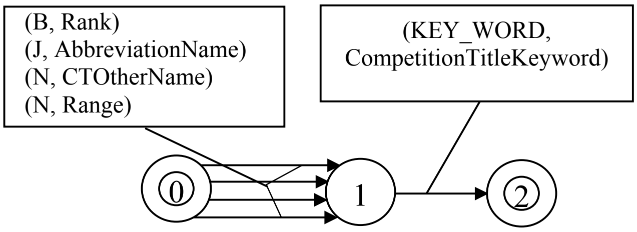


Figure 6.4 An Automaton's Graph

Definition 6.2: Finite-state cascades (FSC) is a *n*-tuple NM = (M_1, M_2, …, M_n), where *n* is a level number of FSC. M_1, M_2, …, M_n are *n* FSAs, in which M_l and M_n correspond to the lowest-level and highest-level FSAs of FSC respectively.

Definition 6.3: The automatic construction of FSC is to automatically transform ordinal recognition rule sets into NM, that is, each rule set is transformed into a M_i from low to high level.

According to the *graph theory* (Aho et al., 1983), we proposed an algorithm for automatically constructing FSC. The main ideas concerning optimized construction are: (i) under the condition of correct construction, the redundancy of edges and states should be minimized; (ii) for the sake of reducing the complexity of FSC, the self-edges that begin and end at the same state and the cycles between two states are not considered. Based on such a construction strategy, we can ensure the *correctness* of FSC and also enhance the *efficiency* of NE identification.

Definition 6.4: In the construction algorithm, in order to effectively store nodes and edges of FSC, we use four adjacent matrices, *POS matrix* (*POSM*), *POS index matrix* (*POSIM*), *semantic index matrix* (*SIM*) and *semantic constraint matrix* (*SCM*), which are used as data structure, replacing the transition function δ:

(i) POSM is used to store POS tags from recognition rules between two states:

POSM = $\{(q_1, q_2, S_1) \mid q_1, q_2 \in Q; S_1 \subseteq S;$ q_1 is a starting state, q_2 is an arriving state, S_1 is a set of all accepted POS tags between q_1 and q_2, and S is the set of all POS tags.$\}$

(ii) POSIM provides the line address pointer to SIM, which is related to POS tags between two states:

POSIM = $\{(q_1, q_2, la_{SIM}) \mid q_1, q_2 \in Q; la_{SIM} \in LA_{SIM};$ q_1 is a starting state, q_2 is an arriving state, la_{SIM} is a line address of SIM; LA_{SIM} is the set of all line addresses of SIM.$\}$

(iii) SIM indicates the position of semantic tags associated with each POS tag in SCM:

SIM = $\{(la_{SIM}, s_1, la_{SCM}) \mid la_{SIM} \in LA_{SIM}; la_{SCM} \in LA_{SCM}; s_1 \in S_1;$ la_{SCM} is a line address of SCM; LA_{SCM} is the set of all line addresses of SCM; s_1 is a POS tag of S.$\}$

(iv) SCM saves the semantic tags for each POS tag in POSM:

SCM = $\{(la_{SCM}, s_2, bool) \mid la_{SCM} \in LA_{SCM}; s_2 \in S_2; bool = 1;$ s_2 is one of the accepted semantic tags corresponding to s_1 in SIM.$\}$

Among four adjacent matrices, POSM and SCM are kept as POS and semantic tags respectively; while POSIM and SIM only are two serial bridges joining POSM and SCM. Therefore, using these matrices, we can represent all information from recognition rule sets in FSC, regardless of the multi-POS tags between two states or the POS tags that have multi-semantic tags.

In order to distinctly describe FSC construction procedure, we would like to give the following primary construction algorithm by more detailed pseudo codes:

Construction Algorithm:

```
main()

input level_size;
for (level_index = 1 to level_size)
    initialize recognition_rule();
    input a NE recognition rule set into recognition_rule();
    initialize posm(), posim(),  sim(), and scm();
```

```
state_index ← 0;
initial_state ← state_index;
rule_set_size ← get_rule_size(recognition_rule());
for (rule_index = 1 to rule_set_size)
    {get a recognition rule containing POS and SEM tags[22]}
    pos_rule() ← recognition_rule(rule_index, POS);
    sem_rule() ← recognition_rule(rule_index, SEM);
    if (rule_index == 1)
        {add the first recognition rule in the recognizer}
        pos_tag_size ← get_tag_size(pos_rule());
        for (pos_tag_index = 1 to pos_tag_size)
            pos_tag ← pos_rule(rule_index, pos_tag_index);
            add pos_tag into posm(state_index, state_index+1);
            add sem_rule(rule_index, pos_tag) into scm() indexed by
             posim() and sim();
            state_index ← state_index + 1;
            final_state ← state_index;
    else
        add_rule(pos_rule(), sem_rule(), rule_index , initial_state,
         final_state);
{put a constructed recognizer into a level of the FSC}
fsc_posm(level_index, , ) ← posm();
fsc_posim(level_index, , ) ← posim();
fsc_sim(level_index, , ) ← sim();
fsc_scm(level_index, , ) ← scm();
```

```
add_rule(pos_rule(), sem_rule(), rule_index, initial_state, final_state)

state_i ← initial_state;
state_max ← get_state_size(posm());
pos_tag_index ← 0;
pos_tag_size ← get_tag_size(pos_rule());
{examine whether the same rule already exists in this recognizer}
```

[22] A comment on the following pseudo codes is given within the braces.

```
if (pos_rule(rule_index, ) not in posm() and sem_rule(rule_index, ) not in
        scm())
    pos_tag_index ← pos_tag_index + 1;
    pos_tag ← pos_rule(rule_index, pos_tag_index);
    sem_tags() ← sem_rule(rule_index, pos_tag);
    state_j ← state_i + 1;
    while (pos_tag_index <= pos_tag_size)
        {exclude a self-edge for a single state}
        if (state_j == state_i)
            state_j ← state_j + 1;
            if (state_j > state_max)
                {add a new state}
                state_max ← state_max + 1;
                {exclude the possibility of cycle's emergence between $state_i$
                 and $state_j$}
                while (exist_circle(pos_tag, sem_rule(rule_index, pos_tag),
                        state_i, state_j))
                    state_j ← state_j + 1;
                    if (state_j > state_max)
                        state_max ← state_max + 1;
                {for the sake of ensuring correctness, add an edge of POS and
                 SEM, the in-degree of starting state and the out-degree of
                 arriving state must be less than or equal to 1.}
                while (in_degree(state_i) > 1 or out_degree(state_j) > 1)
                    if (same_pos(pos_tag, posm(state_i, state_j)) and
                            same_sem(sem_tags(pos_tag, ), scm()))
                        break;
                    else
                        if (in_degree(state_i) > 1)
                            {backtrack to find another state, in which an edge
                             of POS and SEM tags can be added.}
                            if (stack() != empty)
                                state_j ← pop();
                                state_i ← pop();
                                sem_ tags() ← pop();
                                pos_tag ← pop();
                pos_tag_index ← pos_tag_index - 1;
            state_j ← state_j + 1;
            if (state_j > state_max)
                state_max ← state_max + 1;
{examine whether both POS tags are identical, but their SEM
  tags conflict.}
if (not conflict(pos_tag, posm(state_i, state_j)) and not
     conflict(sem_tags(pos_tag, ), scm()))
    if (state_j == final_state and pos_tag_index ==
       pos_tag_size)
        {if a rule is successfully added}
        if (not same_pos(pos_tag, posm(state_i, state_j)) and
            not same_sem(sem_tags(pos_tag, ), scm()))
            add pos_tag into posm(state_i, state_j);
            add sem_ tags(pos_tag, ) into scm() indexed by
             posim() and sim();
        if (same_pos(pos_tag, posm(state_i, state_j)) and
            not same_sem(sem_tags(pos_tag, ), scm()))
            add sem_ tags(pos_tag, ) into scm() indexed by
             posim() and sim();
        successful_match();
        break;
    else
        if (state_j != final_state)
            if (not same_pos(pos_tag, posm(state_i, state_j))
                or not same_sem(sem_tags(pos_tag, ), scm()))
                {temporarily push the related information of a
                  edge of POS and SEM into the stack}
                 push(pos_tag);
                 push(sem_tags(pos_tag, ));
                 push(state_i);
                 push(state_j);
            {get a new rule of POS and SEM}
             pos_tag_index ← pos_tag_index + 1;
             pos_tag ← pos_rule(rule_index, pos_tag_index);
             sem_tags() ← sem_rule(rule_index, pos_tag);
```

```
64)                     state_i ← state_j;
65)                     state_j ← initial_state + 1;
66)                 else
67)                     state_j ← state_j + 1;
68)                     if (state_j >   state_max)
69)                         state_max ← state_max + 1;
70)         else
71)             state_j ← state_j + 1;
72)             if (state_j >   state_max)
73)                 state_max ← state_max + 1;
```

```
 1) successful_match()
 2)
 3) while (stack() != empty)
 4)     state_j ← pop();
 5)     state_i ← pop();
 6)     sem_tags() ← pop();
 7)     pos_tag ← pop();
 8)     if (not same_pos(pos_tag, posm(state_i, state_j)) and not
           same_sem(sem_tags(pos_tag, ), scm()))
 9)         add pos_tag into posm(state_i, state_j);
10)         add sem_ tags() into scm() indexed by posim() and sim();
11)     if (same_pos(pos_tag, posm(state_i, state_j)) and not
12)         same_sem(sem_tags(pos_tag, ), scm()))
13)         add sem_ tags() into scm() indexed by posim() and sim();
```

Notice that it is important that the *correct construction condition* in the procedure of adding a new POS tag's edge must be met, for example, whether its corresponding semantic tags *conflict* with the semantic tags of a existing POS tag edge in the NE recognizer (in the line 42 in add_rule()). Otherwise, adding this edge is *given up* (from line 71 to line 73 in add_rule()).

Broadly speaking, this algorithm principally contains a *heuristic* state search mechanism, that is, it is not an *exhaustive* search algorithm. For example, if the in-degree of a starting state is greater than 1, it doesn't go on finding the arriving state. Instead, it pops the last edge in the top of the stack and renews a search for a suitable arriving state of this edge (from the line 32 to line 37 in add_rule()).

The construction algorithm also is a rule-driven algorithm and only relies upon the representation of rules. Therefore, the constructed FSC are flexible and maintainable in that it is *easy* to *change* the size of POS and semantic tags, and *easy* to *add*, *modify* or *delete* the recognition rules. Additionally, because this algorithm can be applied to establish different recognition levels, it is also *easy* to *expand* the NE recognizers in FSC for *new NEs*.

In the next subsection, an example of the recognizer construction will be elaborated.

6.2.4 An Example of the Constructed Recognizer

Adopting the above algorithm, the FSC in our system can be automatically constructed by the NE recognition rule sets and consists of three recognition levels. Each level has an *NE recognizer*, that is, TN, CT and PI recognizer in turn. As an example, it shows how the following CT recognition rule set is used to construct a *deterministic* finite automaton, that is, a CT recognizer, using the data structure and algorithm represented in the last subsection.

A recognition rule set for CT, which is composed of 15 rules, is a subset of the practical recognition rule set for CT. It is shown as follows:

rule$_1$: CT → B + KEY_WORD | Rank + CompetitionTitleKeyword

rule$_2$: CT → J + KEY_WORD | AbbreviationName + CompetitionTitleKeyword

rule$_3$: CT → N + KEY_WORD | AbbreviationName + CompetitionTitleKeyword | CTOtherName + CompetitionTitleKeyword | Range + CompetitionTitleKeyword

rule$_4$: CT → N1 + KEY_WORD | CountryName + CompetitionTitleKeyword

rule$_5$: CT → N7 + KEY_WORD | CityName + CompetitionTitleKeyword | ContinentName + CompetitionTitleKeyword | CountryName + CompetitionTitleKeyword

rule$_6$: CT → J + B + KEY_WORD | AbbreviationName + Rank + CompetitionTitleKeyword

rule$_7$: CT → N + J + KEY_WORD | CTOtherName + AbbreviationName + CompetitionTitleKeyword

rule$_8$: CT → N1 + N + KEY_WORD | CountryName + CTOtherName + CompetitionTitleKeyword

rule$_9$: CT → N7 + N + KEY_WORD | CityName + CTOtherName + CompetitionTitleKeyword | ContinentName + CTOtherName + CompetitionTitleKeyword | ContinentName + TNOtherName + CompetitionTitleKeyword | CountryName + CTOtherName + CompetitionTitleKeyword

rule$_{10}$: CT → N + M + QT + KEY_WORD | ProductName + Rank + AlphabeticalString + CompetitionTitleKeyword | CTOtherName + Rank + AlphabeticalString + CompetitionTitleKeyword

rule$_{11}$: CT → N7 + N + N + KEY_WORD | CountryName + CTOtherName + CTOtherName + CompetitionTitleKeyword | ContinentName + TNOtherName + CTOtherName + CompetitionTitleKeyword | CTOtherName + CTOtherName + CTOtherName + CompetitionTitleKeyword

rule$_{12}$: CT → A + N + J + B + KEY_WORD | Range + CTOtherName + AbbreviationName + Rank + CompetitionTitleKeyword

rule$_{13}$: CT → A + N + N + J + KEY_WORD | Range + CTOtherName + CTOtherName + AbbreviationName + CompetitionTitleKeyword

rule$_{14}$: CT → A + N + N + M + QT + KEY_WORD | Range + CTOtherName + CTOtherName + Rank + AlphabeticalString + CompetitionTitleKeyword

rule$_{15}$: CT → M + DT + A + N + J + B + KEY_WORD | NumericalString + Quantifier + Range + CTOtherName + AbbreviationName + Rank + CompetitionTitleKeyword

where the POS tags A, B, DT, J, KEY_WORD, M, N, N1, N7, and QT represent an adjective, a discrimination word, a date or time word, an abbreviated word, a keyword of competition title, a numeral, a proper noun, a transliterated noun, and an alphabetical string separately. In the semantic tags, Rank and Range mean the competition rank and range, such as super (rank), woman (range), etc.

Depending on the construction algorithm, when the input sequence of recognition rules is from rule$_1$ to rule$_{15}$ in turn, the contents of four matrices are listed in Figure 6.5 after the recognizer is established.

For this example, we should explain some critical points in the procedure of building four adjacent matrices:

- The first rule of a recognition set is directly added to the matrices. At the same time, the initial and final states of a recognizer are determined.
- If there are multiple POS tags between two states in POSM, we use binary bits to record corresponding POS tags.
- Note that there is a two-stage index by POSIM and SIM. The data in POSIM indicate the first-stage index, which is an address of the second-stage index in SIM. By this index, the address of stored semantic tags in SCM can be fixed. These semantic tags correspond to the POS tags in POSM, e.g., at first, the POS tag “B” from the first recognition rule is put into POSM(0, 1). Here 0

and 1 represent state 0 and state 1 respectively. Then the address counter$_1$ assigns an address “1” to POSIM(0, 1). Thus it fulfils the first-stage index. According to this index, the address “1” of SCM as the second-stage index is set to SIM(1, B) by the address counter$_2$. Moreover, through this index, the symbol “1” is marked in SCM(1, Rank). Now, the storage of POS tag “B” and its corresponding semantic tag “Rank” has been accomplished.

POS Adjacent Matrix (POSM)

$$
\begin{array}{c|cccccccccccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
\hline
0 & & \text{B/J/N/N1/N7} & & \text{J} & \text{N} & \text{N1/N7} & & \text{N7} & \text{A} & & & & \text{M} & & & \\
1 & & & \text{KEY\_WORD} & & & & & & & & & & & & & \\
2 & & & & & & & & & & & & & & & & \\
3 & & \text{B} & & & & & & & & & & & & & & \\
4 & & \text{J} & & & & & \text{M} & & & & & & & & & \\
5 & & \text{N} & & & & & & & & & & & & & & \\
6 & & \text{QT} & & & & & & & & & & & & & & \\
7 & & & & & & \text{N} & & & & & & & & & & \\
8 & & & & & & & & & & \text{N} & & & & & & \\
9 & & & & \text{J} & & & & & & & \text{N} & & & & & \\
10 & & \text{J} & & & & & & & & & & \text{M} & & & & \\
11 & & \text{QT} & & & & & & & & & & & & & & \\
12 & & & & & & & & & & & & & & \text{DT} & & \\
13 & & & & & & & & & & & & & & & \text{A} & \\
14 & & & & & & & & & & & & & & & & \text{N} \\
15 & & & & \text{J} & & & & & & & & & & & &
\end{array}
$$

POS Index Adjacent Matrix (POSIM)

$$
\begin{array}{c|cccccccccccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
\hline
0 & & 1 & & 4 & 6 & 8 & & 12 & 15 & & & & 24 & & & \\
1 & & & 2 & & & & & & & & & & & & & \\
2 & & & & & & & & & & & & & & & & \\
3 & & 3 & & & & & & & & & & & & & & \\
4 & & 5 & & & & & 10 & & & & & & & & & \\
5 & & 7 & & & & & & & & & & & & & & \\
6 & & 9 & & & & & & & & & & & & & & \\
7 & & & & & & 11 & & & & & & & & & & \\
8 & & & & & & & & & & 14 & & & & & & \\
9 & & & & 13 & & & & & & & 17 & & & & & \\
10 & & 16 & & & & & & & & & & 19 & & & & \\
11 & & 18 & & & & & & & & & & & & & & \\
12 & & & & & & & & & & & & & & 23 & & \\
13 & & & & & & & & & & & & & & & 22 & \\
14 & & & & & & & & & & & & & & & & 21 \\
15 & & & & 20 & & & & & & & & & & & &
\end{array}
$$

Semantic Index Adjacent Matrix (SIM)

$$\begin{array}{c|cccccccccc}
 & \text{A} & \text{B} & \text{DT} & \text{J} & \text{KEY\_WORD} & \text{M} & \text{N} & \text{N1} & \text{N7} & \text{QT} \\
\hline
1 & & 1 & & 3 & & & 4 & 5 & 6 & \\
2 & & & & & 2 & & & & & \\
3 & & 7 & & & & & & & & \\
4 & & & & 8 & & & & & & \\
5 & & & & 9 & & & & & & \\
6 & & & & & & & 10 & & & \\
7 & & & & & & & 11 & & & \\
8 & & & & & & & & 12 & 13 & \\
9 & & & & & & & & & & 14 \\
10 & & & & & & 15 & & & & \\
11 & & & & & & & 16 & & & \\
12 & & & & & & & & & 17 & \\
13 & & & & 18 & & & & & & \\
14 & & & & & & & 19 & & & \\
15 & 20 & & & & & & & & & \\
16 & & & & 21 & & & & & & \\
17 & & & & & & & 22 & & & \\
18 & & & & & & & & & & 23 \\
19 & & & & & & 24 & & & & \\
20 & & & & 25 & & & & & & \\
21 & & & & & & & 26 & & & \\
22 & 27 & & & & & & & & & \\
23 & & & 28 & & & & & & & \\
24 & & & & & & 29 & & & &
\end{array}$$

Semantic Constraint Adjacent Matrix (SCM)

$$\begin{array}{c|ccccccccccccc}
 & \text{Abbreviation-Name} & \text{Alphabetical-String} & \text{City-Name} & \text{Competition-TitleKeyword} & \text{Continent-Name} & \text{Country-Name} & \text{CTOther-Name} & \text{Numerical-String} & \text{Product-Name} & \text{Ouantifier} & \text{Range} & \text{Rank} & \text{TNOther-Name} \\
\hline
1 & & & & & & & & & & & & 1 & \\
2 & & & & 1 & & & & & & & & & \\
3 & 1 & & & & & & & & & & & & \\
4 & 1 & & & & & & 1 & & & & 1 & & \\
5 & & & & & & 1 & & & & & & & \\
6 & & & 1 & & 1 & 1 & & & & & & & \\
7 & & & & & & & & & & & & 1 & \\
8 & 1 & & & & & & & & & & & & \\
9 & 1 & & & & & & & & & & & & \\
10 & & & & & & & 1 & & 1 & & & & \\
11 & & & & & & & 1 & & & & & & \\
12 & & & & & & 1 & & & & & & & \\
13 & & & 1 & & 1 & 1 & & & & & & & \\
14 & & 1 & & & & & & & & & & & \\
15 & & & & & & & & & & & & 1 & \\
16 & & & & & & & 1 & & 1 & & & & \\
17 & & & & & 1 & 1 & 1 & & & & & & \\
18 & 1 & & & & & & & & & & & & \\
19 & & & & & & & 1 & & & & & & \\
20 & & & & & & & & & & & 1 & & \\
21 & 1 & & & & & & & & & & & & \\
22 & & & & & & & 1 & & & & & & \\
23 & & 1 & & & & & & & & & & & \\
24 & & & & & & & & & & & & 1 & \\
25 & 1 & & & & & & & & & & & & \\
26 & & & & & & & 1 & & & & & & \\
27 & & & & & & & & & & & 1 & & \\
28 & & & & & & & & & & 1 & & & \\
29 & & & & & & & & 1 & & & & &
\end{array}$$

Figure 6.5 An Example to Establish a CT Recognizer Using Four Adjacent Matrices

Figure 6.6 illustrates the constructed CT recognizer which corresponds to the above four adjacent matrices. Note that the semantic tags are *not* depicted due to limited figure space.

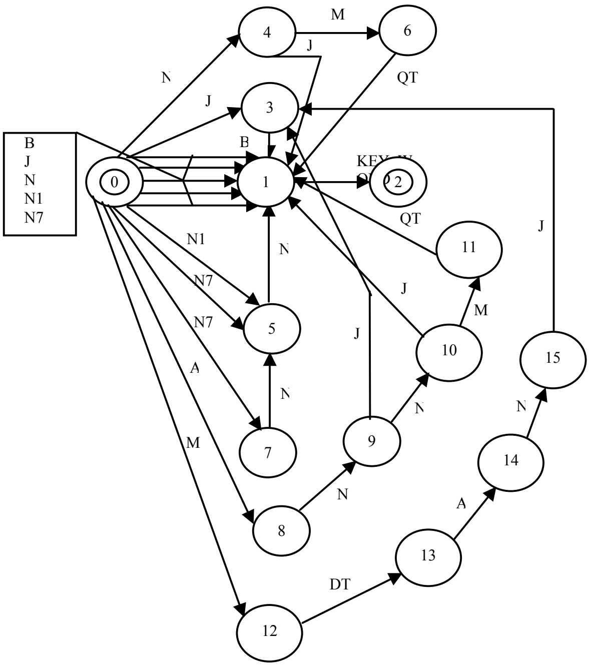


Figure 6.6 An Automatically Constructed CT Recognizer in FSC

6.3 Identification Procedure

6.3.1 Required Resources

The recognition rules deal with not only POS constraints, but also corresponding semantic constraints. Because the input words only include POS tags, before the recognition of the named entity we must provide related semantic information as semantic tags for the words. For this purpose, we construct different name libraries[23] such as continent, country, province (state), city name, and the commonly used club, company, product, abbreviation name (it contains continent, country, province, state, city, club names, etc.), team name keyword, competition title keyword, personal title, etc. in the sports domain. In addition, we establish other libraries associated with domain words, e.g., range (competition range) and rank (competition rank) libraries. These names and words are partly acquired from Sports Ontology and are partly crafted manually from corpus, e.g., *Jie Fang Daily*.

In the next subsection, we describe this algorithm in detail. Subsection 6.3.3 gives an identification example to further specify this algorithm.

[23] Because the required semantic information for NE identification exceeds Sports Ontology's taxonomy, in order to unify data format, we construct these name libraries for a gazetteer to annotating semantic tags and for identifying TN and CT with special constructions.

6.3.2 Identification Algorithm

As presented previously, we know that the construction algorithm serves as an automatic FSC constructor relying upon the recognition rule sets. After FSC is established, however, how can we utilize this constructed FSC to further identify Chinese named entities? In this subsection, a proposed identification algorithm will be explained. This algorithm detects named entities in the order from low-level to high-level recognizers. Each recognizer's output, as an input, is directly sent to its high-level recognizer. Thus, a high-level recognizer can benefit from the outcome of its low-level recognizer. In the following, we elaborate the identification algorithm via the pseudo code:

Identification Algorithm:

```
main()

input fsc_posm(), fsc_posim(), fsc_sim(), fsc_scm(), and level_size;
input a text into text();
input initial_state() and final_state();
sentence_size ← get_sentence_size(text());
for (sentence_index = 1 to sentence_size)
    sentence_words() ← text(sentence_index, WORD);
    sentence_pos_tags() ← text(sentence_index, POS);
    {retrieve different name libraries and determine semantic tag for each word.}
    sentence_sem_tags() ← get_sem_tags(sentence_words());
    for (level_index = 1 to level_size)
        initialize posm(), posim(), sim(), and scm();
        posm() ← fsc_posm(level_index, , );
        posim() ← fsc_posim(level_index, , );
        sim() ← fsc_sim(level_index, , );
        scm() ← fsc_scm(level_index, , );
        word_size ← get_word_size(sentence_words());
        initial_state ← initial_state(level_index);
        final_state ← final_state(level_index);
        state_i ← initial_state;
        state_j ← state_i + 1;
        state_max ← get_state_size(posm());
        word_index ← 1;
        named_entity_begin ← word_index;
```

```
while (word_index <= word_size)
    if (state_j == state_i)
        state_j ← state_j + 1;
        while (state_j > state_max)
            if (stack != empty)
                {backtrack to the last word and its states.}
                state_j ← pop();
                state_i ← pop();
                word_index ← pop();
                state_j ← state_j + 1;
            else
                {renew a detection for a named entity.}
                output sentence_pos_tags(word_index);
                word_index ← word_index + 1;
                named_entity_begin ← word_index;
                state_i ← initial_state;
                state_j ← state_i + 1;
                break;
    if (posm(state_i, state_j) > 0 and  find_same_pos(sentence_pos_
       tags(word_index), posm(state_i, state_j)) and find_same_sem
       (sentence_sem_tags(word_index), scm()))
        if (state_j == final_state)
            {successfully identify a named entity.}
            named_entity_end ← word_index;
            output_named_entity(level_index, named_entity_begin,
             named_entity_end);
            word_index ← named_entity_end + 1;
            state_i ← initial_state;
            state_j ← state_i + 1;
        else
            {recognize a candidate word for a named entity.}
            push(word_index);
            push(state_i);
            push(state_j);
            {get the next word and go on matching this word.}
            word_index ← word_index + 1;
```

```
                state_i ← state_j;
                state_j ← initial_state + 1;
        **else**
            state_j ← state_j + 1;
            **while** (state_j >  state_max)
                **if** (stack != empty)
                    state_j ← pop();
                    state_i ← pop();
                    word_index ← pop();
                    state_j ← state_j + 1;
                **else**
                    output sentence_words(word_index);
                    output sentence_pos_tags(word_index);
                    output sentence_sem_tags(word_index);
                    word_index ← word_index + 1;
                    named_entity_begin ← word_index;
                    state_i ← initial_state;
                    state_j ← state_i + 1;
                    **break;**
```

```
output_named_entity(level_index, named_entity_begin, named_entity_end)

**for** (named_entity_index = named_entity_begin **to** named_entity_end)
    **if** (named_entity_index < named_entity_end)
        sentence_pos_tags(named_entity_index) ← get_named_entity_tag
         (level_index);
    output sentence_words(named_entity_index);
    output sentence_pos_tags(named_entity_index);
    output sentence_sem_tags(named_entity_index);
```

where (i) the data in fsc_posm(), fsc_posim(), fsc_sim(), fsc_scm() are set by the FSC construction algorithm; (ii) text() involves the sentences to be recognized; (iii) level_index is assigned from low (1) to high (level_size) level; (iv) named_entity_begin and named_entity_end are two positions which separately denote the begin and end of a named entity in a sentence.

Note that we arrange a backtracking mechanism in this algorithm (in line 30 to line 35 and line 64 to line 68). Through this mechanism, the algorithm can detect other means of reaching the final state.

The next subsection describes the identification procedure of the above algorithm.

6.3.3 Analysis of Identification Procedure

For the sake of further explaining the identification algorithm, especially in combination with the example in subsection 6.2.4, we give an example to describe the CT identification procedure with FSC. Figure 6.7 illustrates the NE identification procedure using the three-level recognizers of FSC.

Example 6.4:

An input sentence including words and their POS is shown as follows:

上海|N5|申花|N|队|N|在|P|百事可乐|N|甲|M|Ａ|QT|联赛|N|中|N|击败|V|对手|N|吉林|N5|敖东|N|队|N|。|W|

Shanghai Shenhua Team defeated its opponent, the Jilin Aodong Team, in the Pepsi First A League Matches.

Step 1 (input a sentence and relevant POS and semantic tags):

sentence_words() = [上海 (Shanghai), 申花 (Shenhua), 队 (Team), 在 (in), 百事可乐 (Pepsi), 甲 (First), Ａ, 联赛 (League Matches), 中 (in), 击败 (defeat), 对手 (opponent), 吉林 (Jilin), 敖东 (Aodong), 队 (Team), 。(.)]

sentence_pos_tags() = [N5, N, KEY_WORD, P, N, M, QT, N, N, V, N, N5, N, KEY_WORD, W]

sentence_sem_tags() = [CityName, CompanyName, TeamNameKeyword, Empty, ProductName, Rank, AlphabeticalString, CompetitionTitleKeyword, Empty, Empty, PersonalTitle, CityName, CompanyName, TeamNameKeyword, PunctuationMark]

Step 2 (from position 0 to position 4[24]):

Here we only discuss the CT identification procedure (i.e., level_index = 2). In Figure 6.7 this procedure is illustrated, but in the figure the semantic tags are *omitted* due to limited figure space. In level L_1, which is the output of the recognizer T_1, sentence_pos_tags() = [TN, TN, KEY_WORD, P, N, M, QT, N, N, V, N, TN, TN, KEY_WORD, W] and sentence_sem_tags() are not changed. The recognizer T_2 begins from position 0 in level L_1. However, the first POS tag cannot match any one of the POS tag edges from initial state to other states (see Figure 6.6). Therefore, the POS tag "TN" is simply outputted. Like this POS tag,

[24] The position is indicted in Figure 6.7. Words and their POS tags are located between two positions. The semantic tags are omitted due to limited figure space.

the following three POS tags also have no transition in T_2. All of them are only outputted.

```
L3  TN   TN   K_W   P    CT      CT   CT   K_W   N    V    PI    TN   TN   K_W   W
 T3
L2  TN   TN   K_W   P    CT      CT   CT   K_W   N    V    N     TN   TN   K_W   W
 T2
L1  TN   TN   K_W   P    N       M    QT   K_W   N    V    N     TN   TN   K_W   W
 T1
L0  N5   N    K_W   P    N       M    QT   K_W   N    V    N     N5   N    K_W   W

  上海  申花   队    在  百事可乐  甲   A   联赛   中   击败  对手  吉林  敖东   队   。
 0    1    2    3    4         5    6    7     8    9    10    11    12    13   14   15
```

Figure 6.7 Identification Procedure of FSC

Note: the word between position 0 and 1 (0-1): “Shanghai”; 1-2: “Shenhua”; 2-3: “Team”; 3-4: “in” (the first part); 4-5: “Pepsi”; 5-6: “First”; 6-7: “A”; 7-8: “League Matches”; 8-9: “in” (the second part); 9-10: “defeat”; 10-11: “opponent”; 11-12: “Jilin”; 12-13: “Aodong”; 13-14: “Team”; 14-15: “.”. K_W = KEY_WORD.

Step 3 (from position 4 to position 8):

Then T_2 begins to match POS tag “N” and SEM tag “ProductName” from position 4. Because the SEM tags corresponding to the edge of POS tag “N” between state 0 and state 1 are “AbbreviationName”, “CTOtherName”, or “Range”, obviously there is no match between them. Thus, T_2 continuously seeks other possibilities. When it reaches state 4 and finds a POS tag edge whose SEM tags are “CTOtherName” and “ProductName”, because it doesn’t reach the final state, it only pushes the related information into the stack. Starting from state 4, at first, it detects the POS tag edge is “J”, there is no match; then it discovers another POS tag edge whose POS and SEM tags are “M”, “Rank” respectively. Therefore, it produces a match and still pushes the information into the stack. Going on the next search, only a POS tag edge is available, its POS and SEM tags are “QT” and “AlphabeticalString” separately. A match is produced, and the related information is pushed into the stack. Setting out from state 1, T_2 reaches the final state 2 associated with the named entity CT in the position 8, this POS tag edge matches the POS and SEM tags - “KEY_WORD” and “CompetitionTitleKeyword”. As a result, there is a transition from position 4 to 8 which outputs a “CT” at L_2.

Step 4 (from position 8 to position 15):

The recognizer T_2 keeps on matching from the position 8 in level L_1. But all of the remained POS tags cannot coincide with any POS tag edge from initial state to other states, so that there is no new transition in T_2. They are simply outputted.

6.4 Identification for the Named Entities with Special Constructions

In this section, a problem existing in NE identification and its solution will be discussed. As we have mentioned in the previous sections, FSC mainly rely upon the recognition rules. Therefore, if a named entity has a special construction, especially if its construction conflicts with another named entity's, we cannot use FSC to identity it. For instance, sometimes there are TNs or CTs with special constructions in sentences (see Example 5.6 and 5.7). Here, we would like to duplicate these two examples and complement an example with regard to a phrase structure as follows:

Example 6.5: 上视队一球小胜大连。(Shanghai Television Team feebly won Dalian by exactly one ball.)

Example 6.6: 中国取胜瑞典并不容易。(It is not easy that China wants to score a success against Sweden.)

Example 6.7: 沪连之战。(the fight between Shanghai and Dalian.)

Observing the above examples, we note that at least one TN has no keyword 队 (Team) in the sentences and the phrase. After FSC is applied to identify TN, only the first TN 上视队 (Shanghai Television Team) can be identified.

Obviously, it is of benefit to identify such a special construction if some exceptional solutions can be provided. So we propose a strategy adopting the following steps for identifying a TN without a keyword. It is developed as an additional component outside the FSC component:

Step 1: *With the help of domain verbs and their valence constraints in Sports Ontology, determine the examination scope of TN without a keyword*

In Sports Ontology, under *Movement* top-concept category, a number of domain verbs which are related to two team names have been collected. According to these verbs, we can detect where we should further check possible special constructions of TN in sentences and phrases.

Step 2 : *Keep the equity of domain verbs and analyze the constituents of TN candidates*

According to the valence constraints, we examine whether the NE tag and concept of constituents in both sides of a domain verb are in accordance with them, e.g., in Example 6.5, the team name of the left side should be balanced with the team name of the right side in terms of the valence constraints of the domain verb 胜 (win). Moreover, the candidate of the team name without a keyword is checked, because its constituent is a city name (Dalian), it can be as a constituent of team name in terms of TN recognition rules.

Step 3 : *Utilize context clue of TN candidates*

In order to enhance the correctness of the recognition for a TN without keyword, we establish whether there is a TN in the context of the TN candidate, whose name except the keyword is equal to the current TN candidate. As an example, namely, in Example 6.6, depending on the valency constraints, we know that a team name can occur on both sides of the domain verb 取胜 (win victory). In addition, we also know that a country name can be a constituent of team name in terms of recognition rules. Besides these conditions, we further observe the context of these two TN candidates. If there is such a context clue, the candidates are inferred as TNs. Otherwise, we go on recognizing the candidates by Step 4.

Step 4: *Distinguish team name from location name*

Because an LN can be a constituent of a TN, we should distinguish a TN without keyword from an LN. With the help of other constituents (e.g., nouns, propositions, etc.) in a sentence, the differences of both NEs can be distinguished to a certain extent. In Example 6.7, the noun 战 (fight) is an analogy for competition in the sports domain. Therefore, here 沪 (Shanghai) and 连 (Dalian) represent two separate TNs. But sometimes it is still difficult to identity such TNs.

In fact, this strategy plays an auxiliary role to the further improvement of the precision of TN recognition. Of course, because of the difficulties of distinguishing between LN and TN, it only heightens the restricted precision of TN identification (see the next section.).

6.5 Experimental Results and Evaluation

In this section, two concerned experiments arranged for this component are exhibited. The first one we want to present is based on the named entity identification by FSC. In general, these recognized named entities are regular in their entity construction. For example, TN and CT comprehend keywords in their entities. Therefore, we can use recognition rules of named entities to construct FSC and then identify them. The second one deals with the identification for special constructions of named entities, such as TN and CT without keywords.

In the experiments, we utilize the *same* testing set for the error repair component to check the named entity identification. Furthermore, the rule sets provided for TN, CT, and PI recognition have 35, 50, and 20 rules respectively. In Sports Ontology, there are more than 350 domain verbs used for the identification of TN with special constructions.

Three measures, i.e., recall, precision, and F-measure, are applied to evaluate the performance of this component. The former two measures are defined in the formulas 6.4 and 6.5, the definition of the last one is the same formula as formula 4.3 (see Chapter 4).

$$\text{Recall} = \frac{\text{correctly recognized named entity number}}{\text{total named entity number}} \qquad (6.4)$$

$$\text{Precision} = \frac{\text{correctly recognized named entity number}}{\text{recognized named entity number}} \tag{6.5}$$

where the total named entity number in the formula 6.4 is manually obtained.

The first experimental results are illustrated in Table 6.1, 6.2, and 6.3. The three measures (*total average recall*, *precision*, and *F-measure*) have achieved 83.38%, 82.79%, and 83.08% separately. In addition, by comparison with these tables, we observe that the performance of six types of named entities has been *manifestly* improved: the total average recall is increased from 57.55% to 83.38%; the total average precision has also increased from 65.16% to 82.79%; based on the both measures, the total average F-measure is relevantly enhanced from 61.12% to 83.08%.

	PN	**DT**	**LN**	**TN**	**CT**	**PI**	**Total Average**
Without Error Repair	37.20	69.45	48.95	38.55	66.05	85.10	57.55
With Error Repair	74.40	93.30	70.95	81.95	86.65	93.00	83.38

Table 6.1 Recall Comparison

	PN	**DT**	**LN**	**TN**	**CT**	**PI**	**Total Average**
Without Error Repair	30.80	92.90	51.95	65.15	71.40	78.75	65.16
With Error Repair	67.10	96.80	80.25	76.35	87.65	88.60	82.79

Table 6.2 Precision Comparison

	PN	**DT**	**LN**	**TN**	**CT**	**PI**	**Total Average**
Without Error Repair	33.70	79.48	50.41	48.44	68.62	81.80	61.12
With Error Repair	70.56	95.02	75.31	79.05	87.15	90.75	83.08

Table 6.3 F-measure Comparison

In Table 6.4, the result of the second experiment is enumerated: the average recall for TN without keyword (86.15%) has exceeded the average recall of TN (81.95%) in Table 6.1; the average recall, precision, and F-measure of CT without keyword (97.78%, 97.73%, and 97.75%) have also exceeded the average recall, precision, and F-measure of CT (86.65%, 87.65%, and 87.15%) in Tables 6.1 to 6.3. Overall, these results specify that the identification of named entities with special constructions has reached a good level. But we notice that the average precision of TN *only* attains 66.07%. For that we analyze the error reasons for the identification of TN without keyword: among 19 errors there are 17 errors of wrong identification of LN and 2 errors of imperfect identification of TN. That is to say, Step 4 of the identification strategy in Section 6.4 should be further improved.

	Total Number	**Total Recognized Number / (Total Error Number)**	**Average Recall**	**Average Precision**	**Average F-measure**
TN Without Keyword	65	56 / (19)	86.15	66.07	74.79
CT Without Keyword	45	44 / (1)	97.78	97.73	97.75

Table 6.4 Identifying Performance for TN and CT with Special Constructions

6.6 Discussion

In this chapter, an approach to FSC-based Chinese NE identification has been elaborated. The motivations for choosing FSC as a Chinese NE identifying mechanism are mainly:

- sometimes a Chinese NE is identified in terms of its context, in other words, it requires a sequence between different NE identifications (maybe such a requirement also exists in other languages). For instance, a number of recognition rules for PI depend on the recognition results for PN and TN (see Appendix D.3). FSC's multi-level mechanism can satisfy this requirement;
- each recognizer in FSC corresponds to a kind of recognized NE. Thus, it is *easy* to produce recognition rule modules and maintain them. In a sense, it *reduces* repetition and redundancy of recognition rules. On the other hand, such arrangement of recognition rules may cut down on *ambiguity* in recognition rules;
- FSC are fast and robust. Furthermore, they have better parsing performance.

Compared to other FSC-based shallow parsers (FASTUS, GATE, SproUT, and CCSP), our shallow parser for NE identification has the following advantages:

- The regular expressions used to construct FSC are allowed to represent complex constraints, that is, use *multiple* constraint symbols rather than *atomic* constraint symbols. Thus, it extends the original definition of FSA, and makes FSC more suitable to practical applications;

- Because the parser is based on FSC which is automatically constructed, it is *more* flexible and maintainable;
- It is *easy* to use the parser for rule developers. They don't need to concern FSC's construction modes, that is to say, it is transparent for them how to construct FSC;
- In addition to the above advantages, the strategy to identifying NE with special constructions is an assistant means giving the solutions for the *special* linguistic phenomena which can be not processed by FSC.

In (Chen et al., 1998), they adopted different types of information from different levels of Chinese text to extract named entities, including character conditions, statistic information, titles, punctuation marks, organization and location keywords, speech-act and locative verbs, cache and n-gram model. Their NTU system has been evaluated in MET-2. The F-measures P&R, 2P&R and P&2R were 79.61%, 77.88% and 81.42%, respectively. Zhang and Zhou. (2000) viewed the entire problem as a series of classification problems and employed memory-based learning (MBL) to resolve them. Their system can extract two named entities (personal and organization name). The performance (recall and precision) for them is shown as follows: *personal name* (86.30%; 83.20%); *organization name* (73.40%; 89.30%). Bontcheva et al., (2003) reported their experimental result from ANNIE System (A Nearly-New IE System) under GATE. The performance (recall and precision) for English NE identification is demonstrated in the following: *address* (81%; 81%); *date* (77%; 67%); *location* (96%; 88%); *money* (47%; 82%); *organization* (39%; 75%); *percent* (82%; 100%); *person* (78%; 68%); and *overall* (67%; 82%). Contrasting these results, they are roughly comparable to the performance of our Chinese IE identification component.

Chapter 7

Named Entity Relation Identification

7.1 Overview

In this chapter, I propose a learning and identifying approach for named entity relations (NERs) called *positive and negative case-based learning and identification* (*PNCBL&I*). The learning in this approach belongs to supervised statistical learning (Nilsson, 1996). Actually, it is a variant of memory-based learning (Stanfill and Waltz, 1986; Daelemans, 1995; Daelemans et al., 2000). This approach pursues the improvement of the identification performance for NERs through simultaneously learning two *opposite* cases (NER and non-NER patterns), automatically selecting *effective multi-level* linguistic features from a predefined feature set for each NER and non-NER, and *optimally* achieving an identification tradeoff.

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 weighting* serve as a proportion determination of feature's degree of importance for identifying NERs against non-NERs. Subsequently, *identification thresholds* can also be determined.

One of the differences from memory-based learning is to transform a case's representation into an NER (or non-NER) pattern that depicts the relationships between the NER (or the non-NER) and its features. In the NER and non-NER patterns, sentence group, NER (or non-NER) expression, and NER (or non-NER) features are enumerated respectively. Thus, it is *easy* to comprehend and access features (because of a great number of feature similarity calculations). Another difference is to add non-NER patterns as another kind of cases. Such cases make it possible to realize negative case-based learning[25] and confirm the correctness of NER identification from the *opposite* side. Therefore, the fundamental resources related to the learning and identification are composed of these two kinds of cases, i.e., NER and non-NER patterns.

Let us compare different functions of positive and negative case. Positive case as a primary identification resource gives NER references, i.e., NER patterns, for new cases. When a new case discovers that the NER environment of a positive case in a sentence group is very similar to its own, the corresponding NER candidate in the new case can be thought of as the same kind of NER. But if a new case with two named entities cannot seek NER

[25] Because if we don't consider negative cases, we don't learn the sentences that *only* contain negative cases at all. That is to say, the cases in these sentences are ignored.

reference, we can by no means draw a conclusion that there is no NER between these two named entities, because we have no *proof* to support such a conclusion. Now negative cases help us supply this proof. If this new case finds a similar negative case for these two named entities, it means that their NER doesn't exist.

However, we may be confronted with the problem that an NER candidate in a new case matches more than one positive case, or at the same time, both positive and negative cases. In such situations, we have to employ a *vote* to decide which existing case environment is more similar to the new case. Thereby, a decision strategy is employed that establishes which NER is chosen or whether the NER exists.

In addition, a number of *special circumstances* should be also considered during the NER identification. One is the *relation conflict* that means there is a contradictory identification result. In order to resolve this problem, we compare the similarity computational results between contradictory relations to decide which one is a correct relation. The other is the *relation omission* - sometimes we may find that some NERs should be in the identified relation set in terms of the relationships among NERs, but they are not involved. The inference of NERs is a strategy to enlarge the identified NER set. By inference, the missing relations can be supplemented.

Figure 7.1 illustrates the architecture of the component for the NER identification in our system. In the figure, the dotted line indicates the flow process of the training texts; while the solid line represents that of the testing texts. In addition, the *related learning results* mean selected features (GCFs or ICFs), corresponding feature weights, and identification thresholds for different NERs or non-NERs.

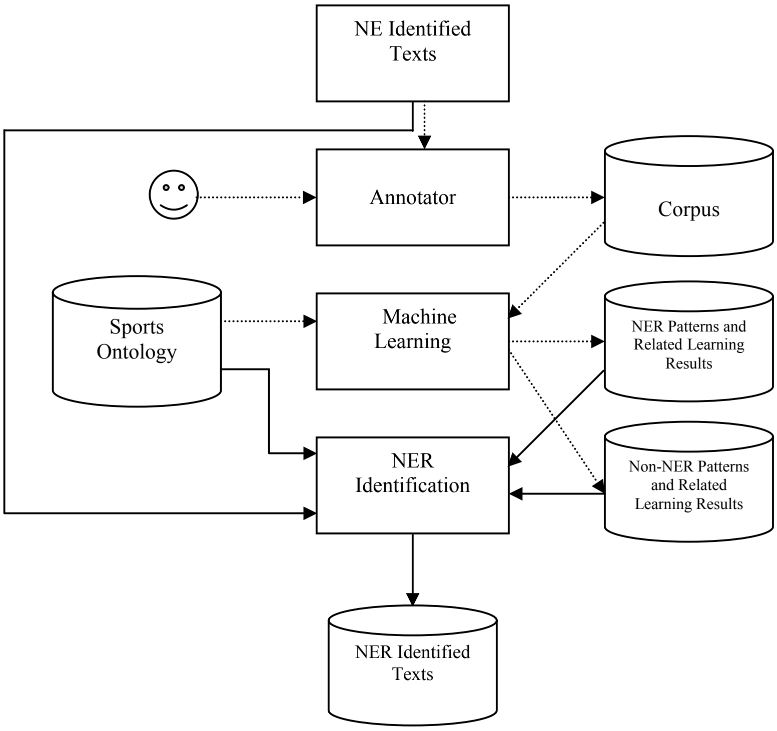


Figure 7.1 Architecture of the Component for the NER Identification

The next section gives all of the relation definitions, an algorithm of the relation annotation and a concrete instance of an annotated sentence. In Section 7.3, the PNCBL machine learning approach using NER and non-NER patterns is described. Then in Section 7.4 a NER identification approach that is based on the above learning results is proposed. After that, in order to specify the NER identification performance of this approach, we give experimental results and compare learning and identification results with or without negative cases in Section 7.5. Finally, we summarize the advantages of our approach and compare our approach with other related methods in Section 7.6.

7.2 Relation Annotation

7.2.1 Definition of Relations

An NER may be a modifying/modified, dominating/dominated, combination, collocation or even cross-sentence constituent relationships between named entities, it is an important *semantic* relationship within a sentence or between sentences.

Mitchell et al. (2002) have defined five types of NERs for the relation annotation in *ACE* (*Automatic Content Extraction*) Program. Moreover, every type has its subtypes. Considering the distribution of different kinds of NERs, we define 14 different NERs based on six identified NEs in the sports domain as follows:

Definition 7.1 (NER Definition):

1) *Person − Team (PS_TM)*

   The *PS_TM* relation specifies the membership of a person in a sports team.

2) *Person − Competition (PS_CP)*

   This relation is about the participation of a person in a sports competition.

3) *Person − City / Province / Country (PS_CPC)*

   Concerns the origin location of a person.

4) *Person − Identification (PS_ID)*

   This relation illustrates a person’s name and her/his identity in a sports team or other occasions.

5) *Home Team − Visiting Team (HT_VT)*

   The relation shows the *home* and *visiting* team name in a sports competition.

6) *Winning Team − Losing Team (WT_LT)*

   The *WT_LT* relation indicates the *winning* and *losing* team name in a sports match.

7) *Draw Team − Draw Team (DT_DT)*

   If two sports teams draw a match, this relation gives the names of these two teams.

8) *Team − Competition (TM_CP)*

   If a sports team takes part in a competition, the relation records the team name and the competition title.

9) *Team − City / Province / Country (TM_CPC)*

   This relation specifies where a sports team comes from.

10) *Identification − Team (ID_TM)*

    The *ID_TM* relation is with respect to the identity of a person which a sports team has.

11) *Competition − Date (CP_DA)*

    It gives the staged date for a sports competition.

12) *Competition − Time (CP_TI)*

    This relation shows the staged time for a sports competition.

13) *Competition − Location (CP_LOC)*

    The relation indicates the location where a sports match is held.

14) *Location − City / Province / Country (LOC_ CPC)*

    The *LOC_ CPC* relation explains the location ownership.

Some of NERs defined above can correspond to four types of the relations (*ROLE*, *PART*, *AT* and *SOCIAL*) defined by Mitchell et al. The corresponding relationships between two kinds of the relations are shown in Table 7.1. In the table, *PER*, *ORG*, *GPE*, *LOC* are named entity tags, which represent a *person*, an *organization*, a *geo-political region*[26], and a *location* respectively.

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 7.2 (NER General Frame):

[26] This NE includes geographical regions defined by political and / or social groups. See Unified EDT Annotation Guidelines V2.5 (http://www.ldc.upenn.edu/Projects/ACE/PHASE2/Annotation/).

Relation Type in ACE	Subtype (NE Pair) in ACE	Relation Type in the Sports Domain
ROLE	Member (PER-ORG)	PS_TM
	Affiliate-Partner (ORG-ORG)	HT_VT; WT_LT; DT_DT
	Citizen-Of (PER-GPE)	PS_CPC
PART	Part-Of (LOC-LOC)	LOC_CPC
AT	Based-In (ORG-LOC)	TM_CPC
SOCIAL	Other-Professional (PER-PER)	It can be inferred by PS_TM and PS_ID NERs.

Table 7.1 Corresponding Relationships between Two Kinds of Relations

NamedEntityRelation (NamedEntity$_1$, ParagraphSentenceNamedEntityNo$_1$; NamedEntity$_2$, ParagraphSentenceNamedEntityNo$_2$) (7.1)

Example 7.1:

广东宏远队[27]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 (7.1), 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).

As we mentioned above, the NERs indicate semantic relationships between NEs within a sentence or between sentences. 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

[27] The underlining of Chinese words means that a named entity consists of these words.

Taiyangshen Team” by the verb “defeat”. Therefore, “Guangdong Hongyuan Team” and “Guangzhou Taiyangshen Team” are the winning and losing team, respectively.

7.2.2 Relation Annotation

Before stating the PNCBL approach, we begin with the specification of text annotation representation, especially for NERs. From the output results of the system *CHINERS* (Yao et al., 2003), the information with regard to word, POS, NE category/word semantics, and word concept (from HowNet) is available. Using *XML* language (Harold, 1998), we annotate Chinese texts with *ourself-defined* annotation tags. Meanwhile, the relation annotation is carried out by *interactive* mode.

Below, we use a *XML-Schema* defined by *XSDL* (XML-Schema Definition Language)[28] (World Wide Web Consortium, 2001) to specify our text annotation structure.

Definition 7.3 (A Text Annotation Structure Defined by XSDL):

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">


<xsd:annotation>
    <xsd:documentation>
        This is a Chinese text schema for the annotation of named entity relations.
        Copyright 2004. All rights reserved.
    </xsd:documentation>
</xsd:annotation>


<xsd:complexType name = "WordType">
    <xsd:attribute name = "english" type = "xsd:string"/>
          <xsd:attribute name = "pos" type = "xsd:string"/>
          <xsd:attribute name = "semantics" type = "xsd:string"/>
          <xsd:attribute name = "concept" type = "xsd:string"/>
</xsd:complexType>

<xsd:complexType name = "PunctuationType">
    <xsd:attribute name = "pos" type = "xsd:string"/>
</xsd:complexType>


<xsd:complexType name = "NamedEntityType">
    <xsd:sequence>
        <xsd:sequence minOccurs="1" maxOccurs="unbounded">
```

[28] An XML-Schema represents the objects, as well as their properties and relationships for an XML application. XSDL is a description language proposed by the W3C’s Schema Working Group.

```
            <xsd:choice>
                      <xsd:element name = "word" type = "WordType"/>
                      <xsd:element name = "punct" type = "PunctuationType"/>
                  </xsd:choice>
   </xsd:sequence>
        <xsd:sequence>
            <xsd:element name = "relation" minOccurs="0"
               maxOccurs="unbounded"/>
                <xsd:complexType>
                    <xsd:attribute name = "type" type = "xsd:string"/>
                    <xsd:attribute name = "neno" type = "xsd:string"/>
                </xsd:complexType>
        </xsd:sequence>
    </xsd:sequence>
    <xsd:attribute name = "type" type = "xsd:string"/>
    <xsd:attribute name = "neno" type = "xsd:string"/>
      </xsd:complexType>


<xsd:element name = "text"/>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "paragraph" minOccurs="1"
               maxOccurs="unbounded"/>
                      <xsd:complexType>
                          <xsd:sequence>
                        <xsd:element name = "sentence" minOccurs="1"
                           maxOccurs="unbounded"/>
                                   <xsd:complexType>
                                       <xsd:sequence minOccurs="1"
                                          maxOccurs="unbounded">
                                     <xsd:choice>
                                         <xsd:element name = "word" type =
                                          "WordType"/>
                                         <xsd:element name = "named-entity"
                                          type = "NamedEntityType"/>
                                         <xsd:element name = "punct" type =
                                          "PunctuationType"/>
                        </xsd:choice>
                    </xsd:sequence>
                    <xsd:attribute name = "no" type = "xsd:string"/>
                    <xsd:attribute name = "type" type = "xsd:string"/>
                      </xsd:complexType>
              </xsd:sequence>
              <xsd:attribute name = "no" type = "xsd:string"/>
          </xsd:complexType>
      </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

The XML-Schema specifies the components and provides the rules by which XML documents are created, validated and processed. In the above definition, a XML-based text annotation structure is described. It contains "*text*", "*paragraph*", "*sentence*", "*named-entity*", "*relation*", "*word*", "*punct*" tags and their attributes. Notice that because sometimes there are NERs related to the *same* kind of NEs, the subtypes of NEs through NER attributes must be indicated (see Example 7.2). Therefore, the NER annotation tags are enclosed in every NE.

The annotation algorithm and the annotated result for the sentence in Example 7.1 are shown as follows:

Annotation Algorithm:

- Input an NE-identified text (text format) from System CHINERS;
- Transform it to a XML tree;
- Capture possible NERs in the text and confirm them by interactive mode with the user, including the determination of host or visiting team, wining or losing team, time or date, and the ownership of locations;
- Add the tags of confirmed NERs to the XML tree;
- Output XML tree.

Example 7.2 (An Annotated Sentence):

```
<sentence no="2" type="body">
    <named-entity type="TN" neno="2-2-1">
        <word English="Guangdong" pos="N5" semantics="TeamName"
         concept="place/provincial/ProperName/China">
          广东
        </word>
        <word English="unknown word" pos="N" semantics="TeamName"
        concept="Empty">
        宏远
      </word>
      <word English="group/team/a row of people" pos="N"
        semantics="TeamNameKeyword" concept="community/human/mass">
        队
      </word>
      <relation type="VT:HT" neno="2-2-2" />
      <relation type="WT:LT" neno="2-2-2" />
      <relation type="TM:CP" neno="2-1-3" />
</named-entity>
<word English="guest field" pos="N" semantics="Empty"
    concept="place/$invite/*compete/sport">
    客场
</word>
<word English="according to/because of/by means of" pos="P"
    semantics="Empty" concept="{According To}/{cause}/{means}">
    以
</word>
<word English="3" pos="M" semantics="NumericalString"
    concept="qValue/amount/cardinal/mass">
     3
</word>
<word English="compare" pos="P" semantics="Empty"
  concept="CompareTo">
  比
</word>
<word English="0" pos="M" semantics="NumericalString"
  concept="qValue/amount/cardinal/mass">
   0
</word>
<word English="beat/defeat/defeat" pos="V" semantics="Empty"
  concept="defeat">
  击败
</word>
<named-entity type="TN" neno="2-2-2">
      <word English="Guanzhou" pos="N5" semantics="TeamName"
        concept="place/city/ProperName/China">
        广州
      </word>
```

```
                <word English="Apollo" pos="N" semantics="TeamName"
                  concept="attribute/bearing/&AnimalHuman">
                  太阳神
                </word>
                <word English="group/team/a row of people" pos="N"
                  semantics="TeamNameKeyword" concept="community/human/mass">
                  队
                </word>
                <relation type="HT:VT" neno="2-2-1" />
                <relation type="LT:WT" neno="2-2-1" />
                <relation type="TM:CP" neno="2-1-3" />
          </named-entity>
          <punct pos="W">
              。
        </punct>
    </sentence>
```

We have stated that there are NERs related to the *same* kind of NEs, e.g., the governor/dependent of TNs (HT/VT, WT/LT) or the ownership of LNs (LOC/CPC). In order to denote their subtype, we annotate NER tags for each NE. Moreover, by the value of the attribute "*type*" of the NER tag, the type to the *left* of the colon gives the subtype of the NE which encloses this NER tag. After that, the value of the attribute "*neno*" (this is a paragraph-sentence-named entity no.) indicates the position of the NE whose subtype occurs to the *right* of the colon. For example, in Example 7.2, the value of the attribute "type" of NER in the first TN is "VT:HT", but that in the second TN is "HT:VT", which means the subtype of the *first* TN is VT and the *second* TN is HT. In addition, the value of the attribute "neno" in the first TN is "2-2-2" and that in the second TN is "2-2-1", which indicate that the HT is the second TN and VT is the first TN.

With the annotated text including NERs, we can further establish NER identification resources and derive identification references using positive and negative case-based learning. The next section elaborates this new machine learning approach.

7.3 Positive and Negative Case-Based Learning

Although positive and negative case-based learning (PNCBL) is a variant of memory-based learning, unlike memory-based learning, PNCBL does *not simply* store cases in memory but transforms case form into *named entity relation* (*NER*) and *non-named entity relation* (*non-NER*) *patterns*. Additionally, it stores not only *positive* cases, but also *negative* cases. Here, it should be clarified that the negative case we mean is a case in which two or more than two named entities (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. It is different from the negative instance defined in (Craven, 1999). In this definition, a negative instance means that a sentence doesn't describe any relation instances.

During 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, learning results provide an identification reference for the forthcoming NER identification. Figure 7.2 clearly illustrates the learning procedure.

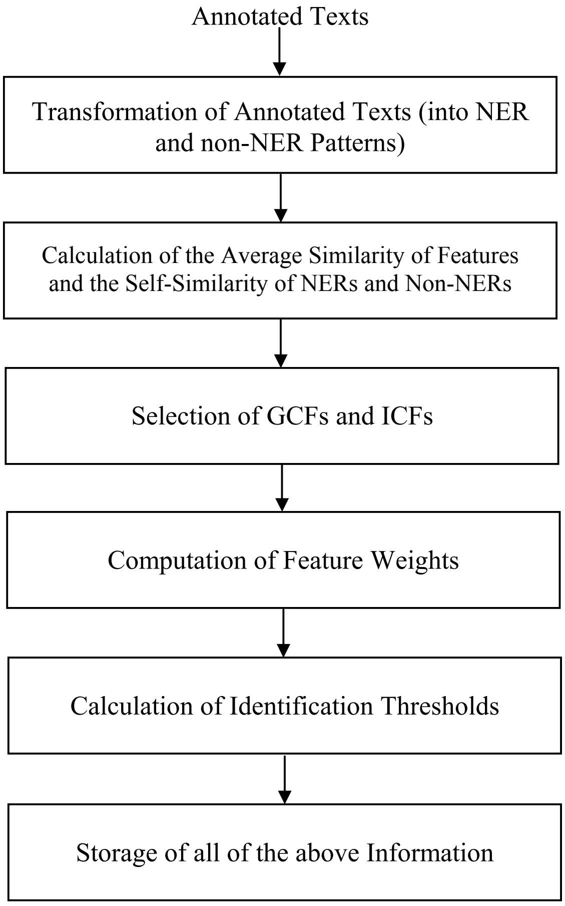

Annotated Texts
Transformation of Annotated Texts (into NER and non-NER Patterns)
Calculation of the Average Similarity of Features and the Self-Similarity of NERs and Non-NERs
Selection of GCFs and ICFs
Computation of Feature Weights
Calculation of Identification Thresholds
Storage of all of the above Information

Figure 7.2 Learning Procedure of PNCBL

In the next subsection, we will depict the definition of NER (non-NER) features, the relationships between relation features and linguistic levels, and the impacts of relation features. Subsection 7.3.2 defines the NER (non-NER) patterns and gives a detailed example of a relation pattern. Then the kernel of this machine learning, a computational approach for the self-similarity of relations, is proposed in Subsection 7.3.3. Finally, the algorithm of PNCBL is described in Subsection 7.3.4.

7.3.1 Definition of Relation Features[29]

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

[29] Note that the relation features are also regarded as the non-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*;
- 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 below, 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 (see Subsection 7.3.2), 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*.

Definition 7.4 (Relation Features):

1) *Sentence Group Type* (*SGTF*)

   The feature SGTF describes the type of a sentence group in which there exists a relation.

2) *Named Entity-Sentence Position* (*NESPF*)

   This feature denotes that the named entities of a relevant relation are located in the same sentence or different sentences.

3) *Named Entity Order* (*NEOF*)

   It indicates the *order* of the named entities of a relevant relation.

4) *Named Entity-Verb Position* (*NEVPF*)

   The feature NEVPF measures 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.

5) *Named Entity Context* (*NECF*)

   It specifies the context of named entities. The context only embodies a word or a character *preceding* or *following* the current named entity.

6) *Verb-Sentence Position* (*VSPF*)

   This feature indicates that the verbs are located in the *same* sentence or *different* sentences in which there is a relevant relation.

7) *Named Entity-Particle POS Order* (*NEPPOF*)

This feature represents the *relative order* between parts-of-speech of particles and named entities. The particles occur within the sentences where the relation is embedded.

8) *Named Entity POS* (*NEPF*)

The feature NEPF shows parts-of-speech of the named entities of a relevant relation.

9) *POS of Context of Named Entity* (*NECPF*)

This feature represents parts-of-speech of the context for the named entities associated with a relation.

10) *Sentence POS* (*SPF*)

It illustrates the *sequence* of parts-of-speech for all sentence constituents within a relation range.

11) *Valence of Verb* (*VVF*)

This feature specifies the *valence* expression of verbs in the sentence(s) where there is a relation embedded.

12) *Named Entity Concept* (*NECTF*)

The feature NECTF gives the *concepts* of the *named entities* of a relevant relation from HowNet.

13) *Verb Concept* (*VCTF*)

It indicates the *concepts* of the *verbs* of a relevant relation from HowNet.

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. The relationships between relation features and corresponding linguistic levels are listed in Table 7.2.

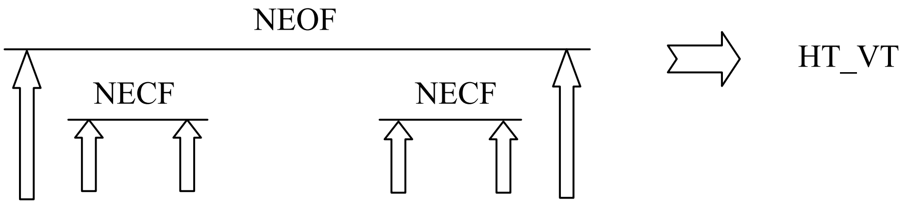
Every feature describes one or more properties of a relation. Through the feature similarity calculation (see Subsection 7.3.3), the quantitative similarity for two relations can be obtained, so that we can further determine whether a candidate relation is a *real* relation. Therefore, the feature definition plays an *important* role for the relation identification. In Example 7.1, the features NECF, NECTF, NEOF, NEVPF, SPF, VCTF and VVF *synthetically* depict these two relations:

Relation Feature	**Linguistic Level**
NECF; NECPF; NEPF	Morphology
NEOF; SPF; SGTF	Grammar
NEPPOF; NESPF; NEVPF; VSPF	Morphology / Grammar
VVF; NECTF; VCTF	Semantics

Table 7.2 Relationships between Relation Features and Linguistic Levels

- *NECF* can capture the noun 客场 (the guest field[30]) and also determine that the *closest* named entity 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 named entities. Thus, another named entity 广州太阳神队 (the Guangzhou Taiyangshen Team) is determined. Therefore, these two features reflect the properties of the relation *HT_VT*.
- *VCTF* can get the *concept* of the verb 击败 (defeat). Based on this concept, *NECTF*, *NEVPF*, *SPF* and *VVF* together can determine the dominating and dominated named entities. That is, 广东宏远队 is a dominating named entity; while广州太阳神队 is a dominated named entity. Obviously, these features can be used for identifying *WT_LT* relation.

Figure 7.3 and 7.4 illustrate the relationships between HT_VT and WT_LT and their related features respectively.

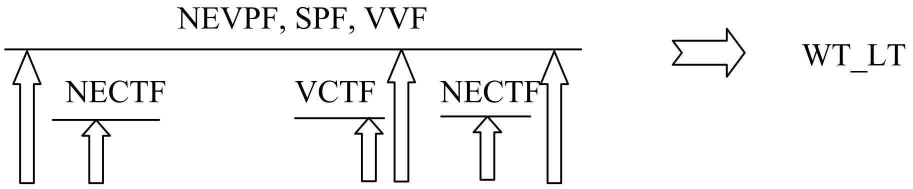


[广东宏远队][客场]以 3 比 0 [击败][广州太阳神队]。

([The Guangdong Hongyuan Team] [defeated] [the Guangzhou Taiyangshen Team] by 3: 0 in [the guest field].)

Figure 7.3 Relationship between the Features and HT_VT Relation

[30] It means that the guest team attends a competition in the host team's residence.



[广东宏远队]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 filed.)

Figure 7.4 Relationship between the Features and WT_LT Relation

7.3.2 Relation and Non-Relation Patterns

7.3.2.1 Relation Pattern

A relation pattern describes the relationships between an NER and its features. In other words, it depicts the *linguistic environment* in which NERs exist. Formally, it consists of the number of a sentence group, relation expressions, sentence content, a sentence type, morphological, grammatical, semantic, and conceptual description items. The following is the formal definition of the relation pattern.

Definition 7.5 (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* ∈ *PINT*. PINT is a set of positive integers. no represents the number of a RP.
- *RE* is a finite set of relation expressions, i.e., RE = {(*rt*, $sneno_1$, $sneno_2$) ∈ *RT* × *SNENO* × *SNENO*}, where *RT* is a finite set of relation types (see Definition 7.1); *SNENO* is a finite set of the numbers corresponding to all the named entities in a sentence group. $sneno_i$ (i = 1,2) is defined as "NE positive integer_1 - positive integer_2", e.g., "NE1-1". Here the first positive integer is a sentence number in the text, and the second positive one is the sequence number of a named entity in this sentence.
- *SC* is a finite set for the words in the sentence group *except for* the words related to named entities. For ∀ *sc* ∈ SC, *sc* = (*sno*, *chiword*, *engexp*, *semantics*, *concept*). Meanwhile, *sno* ∈ PINT, which is a sequence number for the words; *chiword* is a Chinese word; *engexp* is a corresponding English explanation; *semantics* is a semantic constraint given by a gazetteer during named entity identification or a named entity category; *concept* is the word concept from HowNet.
- *sgt* ∈ *SGT*, here *SGT* is a finite set of sentence group types, i.e., *SGT* = {single-sentence, multi-sentences}.
- *NE* is a finite set for *named entities* in the sentence group. For ∀ *ne* ∈ NE, it is a four-tuple, that is, ne = (*sneno*, *sno*, *necat*, *WordSet*), where the definition of *sneno*

is the same as one of $sneno_i$ (see above RE definition); the definition of *sno* is the same as that in SC; *necat* is a category of named entities, i.e., $necat \in NECAT$ = {PN, Date, Time, LN, TN, CT, PI}. *WordSet* is a finite set involving the word sequence numbers in a named entity and the relevant words which are constituents of the named entity. For $\forall$ $ws \in WordSet$, *ws* = (*wno*, *chiword*), here *wno* is a word sequence number in the named entity.

- *NEC* is a finite set that embodies the context of named entities. For $\forall$ $nec \in NEC$, *nec* = (*sneno*, *precont*, *folcont*), where *precont* and *folcont* are the preceding and following context of a named entity separately.

- For $\forall$ $v \in VERB$, *v* = (*sno*, *verb*), *verb* is a Chinese verb. Therefore, VERB is a finite set that includes the sequence numbers of verbs and corresponding verbs.

- *PAR* is a finite set of particles. For $\forall$ $par \in PAR$, *par* = (*sno*, *part*), where *part* is a Chinese particle.

- *NEP* is a finite set of named entities and their POS tags. For $\forall$ $nep \in NEP$, *nep* = (*sneno*, *POSSET*), where *POSSET* is a finite set involving the sequence numbers and relevant POS tags of a name entity. For $\forall$ $ps \in POSSET$, *ps* = (*pno*, *post*). *pno* is a sequence number for the POS tag - *post*. The POS tag set for named entities includes A (Adjective), B (Discrimination), DT (Date or Time), H (Prefix), J (Abbreviated Word), M (Numeral), N (Common Noun), N1 (Special Noun), N5 (Location Name), N7 (Transliterated Name), Q (Quantifier), QT (Alphabetical String), and W (Punctuation).

- *NECP* is a finite set which contains the POS tags of the context for named entities. For $\forall$ $necp \in NECP$, *necp* = (*sneno*, *precontpost*, *folcontpost*), where *precontpost* and *folcontpost* are the POS tag of the preceding and following context for a named entity separately.

- The sentence pattern (*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. SP = $\{(sno, posne) \in PINT \times POSTAG \cup SNENO\}$, where *POSTAG* = {A, B, C, D, DT, F, G, H, I, J, K, L, M, N, N1, N2, N4, N5, N7, P, Q, QT, R, U, V, W}. Here *except for* the above POS tags in the *NEP* definition, C, D, F, G, I, K, L, N2, P, R, U, V represent a conjunction, an adverb, a direction word, a morpheme, an idiom, a suffix, a habitual word, a Chinese personal surname, a proposition, a pronoun, an auxiliary word, and a verb respectively.

- The verb valence (*VV*) is a finite set comprehending *sno* of verbs and its valence constraints (see Argument Case in Subsection 4.4.2) from Sports Ontology. For $\forall$ $vv \in VV$, *vv* = (*vsno*, *VAL*), here *vsno* is defined as "*V_sno*", e.g., "V_8"; *VAL* is a finite set including the valence constraints of this verb.

- *NECT* is a finite set that has the concepts of named entities in a sentence group. For $\forall$ $nect \in NECT$, *nect* = (*sneno*, *necpt*), where $sneno \in SNENO$; *necpt* is a character string. If a named entity consists of multiple words, *necpt* = *concat*($wcpt_1$, "+", $wcpt_2$, "+", … , "+", $wcpt_n$), here *concat* is a function for the connection of character strings; $wcpt_i$ (i = 1, 2, ..., n) is the word concept from HowNet.

- *VCT* is a finite set which gives the concepts of verbs in a sentence group. For $\forall$ $vct \in VCT$, *vct* = (*vsno*, *vcpt*), here *vcpt* is a character string which describes verb concept from HowNet. Note that sometimes there are multiple concepts related to this verb, which are separated by one or more slashes.

In order to further explain the relation pattern, we give the following sentence group as an instance. A detailed relation pattern for this sentence group is shown in Table 7.3.

Example 7.3 (A Sentence Group and its Relation Pattern):

据新华社北京3月26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 occuping the first place of the entire competition.

Relation Pattern

RP = (no, RE, SC, st, NE, NEC, VERB, PAR, NEP, NECP, SP, VV, NECT, VCT), where

no = 34

RE = {(CP_DA, NE1-3, NE1-2), (CP_TI, NE1-3, NE1-4), (TM_CP, NE2-1, NE1-3), (TM_CP, NE2-2, NE1-3), (HT_VT, NE2-2, NE2-1), (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), (5, 电, electricity/cable/telegram, Empty, electricity/letter/letter), (8, 进行, be_on_the_march/march/be_in_progress, Empty, GoForward/GoOn/Vgoingon), (9, 了, empty, Empty, MaChinese), (10, 第, -th, Prefix, aValue/sequence/ordinal), (11, 二, 2/two, NumericalString, qValue/amount/cardinal/mass), (12, 轮, round, Empty, NounUnit/event), (13, 赛事, contest, Empty, fact/compete), (14, 的, empty, AuxiliaryWord, DeChinese), (15, 5, 5, NumericalString, qValue/amount/cardinal/mass), (16, 场, empty, Empty, NounUnit/&RainSnow), (17, 比赛, competition/contest/match, Empty, fact/compete), (18, ，, „ Empty, {punc}), (20, 客场, guest_filed, Empty, place/$invite/*compete/sport), (21, 以, according_to/because_of/by_means_of, Empty, AccordingTo/cause/means), (22, 3, 3, NumericalString, qValue/amount/cardinal/mass), (23, 比, compare, Empty, CompareTo), (24, 0, 0, NumericalString, qValue/amount/cardinal/mass), (25, 击败, beat/defeat/defeat, Empty, defeat), (27, ，, „ Empty, {punc}), (28, 成为, become/turn_into, Empty, become), (29, 唯一, only/sole/one_and_only, Empty, aValue/kind/single), (30, 一, 1/one, NumericalString,

Table 7.3 An Instance of the Relation Pattern

qValue/amount/cardinal), (31, 支, empty, Empty, NounUnit/entity), (32, 两, two, Empty, qValue/amount/cardinal), (33, 战, fight/fight/battle, Empty, fight/fight/military), (34, 全胜, coplete_victory,
Empty, win), (35, 的, empty, AuxiliaryWord, DeChinese), (36, 队伍, troops/contingent/ranks, Empty, army/generic/human/mass), (37, ，, ,, Empty, {punc}), (38, 暂, for_the_moment/for_the_time_being/of_short_duration, Empty, aValue/duration/TimeShort), (39, 居, dwell/live/occupy, Empty, reside/situated), (40, 积分榜, integral_announcement, Empty, account/@record/&result/#compete/sport), (41, 榜首, first_place_in_a_contest/first_place_on_a_list_of_successful_candidates, Empty, attribute/rank/superior/human), (42, 。, ., Empty, {punc})}

st = multi-sentences

NE = {(NE1-1, 3, LN, {(1, 北京)}), (NE1-2, 4, Date, {(1, ３), (2, 月), (3, ２６), (4, 日)}), (NE1-3, 6, CT, {(1, 全), (2, 国), (3, 足球), (4, 甲), (5, Ｂ), (6, 联赛)}), (NE1-4, 7, Time, {(1, 今天)}), (NE2-1, 19, TN, {(1,
广东), (2, 宏远), (3, 队)}), (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), (NE1-3, 电, 今天), (NE1-4, 联赛, 进行), (NE2-1, ，, 客场), (NE2-2, 击败, ，) }

VERB = {(8, 进行), (25, 击败), (28, 成为), (33, 战), (34, 全胜), (39, 居)}

PAR = {(1, 据), (9, 了), (12, 轮), (14, 的), (16, 场), (21, 以), (23, 比), (31, 支), (35, 的), (38, 暂)}

NEP = {(NE1-1, {(1, N5)}), (NE1-2, {(1, M), (2, N), (3, M), (4, N)}), (NE1-3, {(1, A), (2, N), (3, N), (4, M), (5, QT), (6, N)}), (NE1-4, {(1, N)}), (NE2-1, {(1, N5), (2, N), (3, N)}), (NE2-2, {(1, N5), (2, N), (3, N)})}

NECP = {(NE1-1, N, M), (NE1-2, N5, N), (NE1-3, N, N), (NE1-4, N, V), (NE2-1, W, N), (NE2-2, V, W)}

SP = {(1, P), (2, N), (3, NE1-1), (4, NE1-2), (5, N), (6, NE1-3), (7, NE1-4), (8, V), (9, U), (10, H), (11, M), (12, Q), (13, N), (14, U), (15, M), (16, Q), (17, N), (18, W), (19, NE2-1), (20, N), (21, P), (22, M), (23, P), (24, M), (25, V), (26, NE2-2), (27, W), (28, V), (29, A), (30, M), (31, Q), (32, M), (33, V), (34, V), (35, U), (36, N), (37, W), (38, D), (39, V), (40, N), (41, N), (42, W)}

VV = {(V_8, {Agent|fact/compete|CT, -Time|time|DT}), (V_25, {Agent|human/mass|TN, Patient|human/mass|TN}), (V_28, {Agent|human/mass|TN}), (V_33, {Agent|human/sport|PN,

Table 7.3 An Instance of the Relation Pattern (cont.)

Patient|human/mass|TN, Agent|human/mass|TN, Patient|human/sport|PN}), (V_34, {Agent|human/mass|TN, Agent|human/sport|PN, -Patient|human/mass|TN, -Patient|human/sport|PN}), (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), (NE1-3, aValue/range/all+character/surname/human/ProperName+SportTool/fact/exercise+clothing/protect/military+Empty+fact/compete/sport), (NE1-4, time/now/day), (NE2-1, place/provincial/ProperName/China+Empty+community/human/mass), (NE2-2, place/city/ProperName/China+Empty+community/human/mass)}

VCT = {(V_8, GoForward/GoOn/Vgoingon), (V_25, defeat), (V_28, become), (V_33, fight/fight/military), (V_34, win), (V_39, reside/situated)}

Table 7.3 An Instance of the Relation Pattern (cont.)

7.3.2.2 Non-Relation Pattern

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

Definition 7.6 (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. Excepting that, the definitions of other elements are the same as the ones of the relation pattern.

For the content of an NRP, if a sentence group contains both relations and non-relations, except for the element NRE, NRP is identical to RP (but *no* (the number of an NRP) is not required to be equal to one of RP). For example, if we build an NRP for the above sentence group in Example 7.3, the NRE is listed in the following:

NRE = {(CP_LOC, NE1-3, NE1-1), (TM_CPC, NE2-1, NE1-1), (TM_CPC, NE2-2, NE1-1), (HT_VT, NE2-1, NE2-2), (WT_LT, NE2-2, NE2-1), (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)) doesn’t 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.

In Table 7.4, we can observe the corresponding relationships between elements within a NER (non-NER) pattern and NER (non-NER) features. Thus, it is clearly known where the information source for each NER (non-NER) feature is. In other words, the effect of the NER (non-NER) pattern is to provide environment information of NERs (non-NERs), which is associated with *different* linguistic levels. On the other hand, with such an organization model, the relationship between NERs (non-NERs) and their features is *comprehensive* and the data to be processed is also *easy* to access. Note that all of the required data for establishing NER (non-NER) patterns can be obtained from annotated texts and Sports Ontology. Therefore, this procedure can be automatically completed. After that, the NER (non-NER) pattern library should *not* be changed by hand (Daelemans et al., 1999).

In short, NER and non-NER patterns serve as positive and negative case-based learning and identification. In Section 7.3.3, we will present this machine learning approach in detail.

Element of the Relation / Non-Relation Pattern	Feature Category
sgt; RE / NRE	SGTF
RE / NRE	NESPF
RE / NRE	NEOF
NE; VERB; RE / NRE	NEVPF
NEC; RE / NRE	NECF
sgt; VERB; SP; RE / NRE	VSPF
PAR; SP; RE / NRE	NEPPOF
NEP; RE / NRE	NEPF
NECP; RE / NRE	NECPF
NE; SP; RE / NRE	SPF
SP; VV; RE / NRE	VVF
NE; NECT; RE / NRE	NECTF
VCT; SC; SP; RE / NRE	VCTF

Table 7.4 Relationships between Elements of the Relation / Non-Relation Pattern and Relation / Non-Relation Features

7.3.3 Similarity Calculation for Named Entity Relation[31]

In learning, the similarity calculation is a *kernel* measure for feature selection. First, let us look at the definition of *self-similarity* and how to calculate it for the *same* kind of NERs.

Definition 7.7 (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*". In

[31] Analogous to NERs, the computational modes of the similarity can be also used for non-NERs.

Figure 7.5, suppose there are two kinds of relations in a relation set, i.e., r_1 and r_2. Obviously, the concentrative degree of the relation r_1 is tighter than that of the relation r_2.

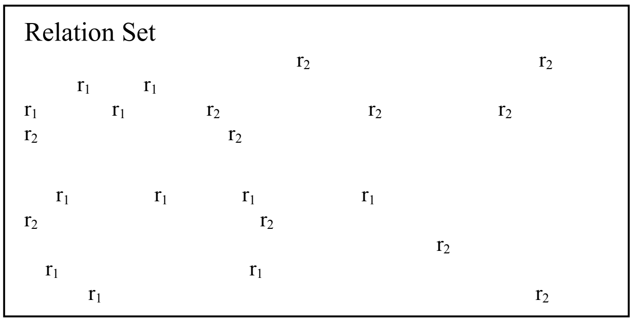


Figure 7.5 Comparison of the Concentration Degree between Two Kinds of Relations

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, that is, self-similarity, is defined as follows:

$$\mathrm{Sim}_{\mathrm{average}}(R(i)) = \frac{\sum\limits_{1\le j,\, k \le m;\, j \ne k} \mathrm{Sim}\,(R(i)_j, R(i)_k)}{\mathrm{Sum}_{\mathrm{relation\_pair}}(R(i)_j, R(i)_k)} \qquad (7.2)$$

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 (see below). *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:

$$\mathrm{Sim}\,(R(i)_j, R(i)_k) = \frac{\sum\limits_{t=1}^{\mathrm{Sum}_f} \mathrm{Sim}\,(R(i)_j, R(i)_k)\,(f_t)}{\mathrm{Sum}_f} \qquad (7.3)$$

$$\mathrm{Sum}_{\mathrm{relation\_pair}}(R(i)_j, R(i)_k) = \begin{cases} 1 & m = 2 \\ \dfrac{m\,!}{(m-2)\,!\, *\, 2\,!} & m > 2 \end{cases} \qquad (7.4)$$

In the formula 7.3, f_t is a feature in the feature set ($1 \leq t \leq 13$). Sum_f is the total number of features. The calculation formulas of *Sim* $(R(i)_j, R(i)_k)$ (f_t) for different features are shown as follows:

1) SGTF

$$\text{Sim}\ (R(i)_j, R(i)_k)\ (\text{SGTF}) = \begin{cases} 1 & \text{if the sentence group types of two relations} \\ & \text{are identical} \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

2) NESPF

$$\text{Sim}\ (R(i)_j, R(i)_k)\ (\text{NESPF}) = \begin{cases} 1 & \text{if the distribution of named entities for two} \\ & \text{relations is the same (either within } \textit{one} \\ & \text{or in } \textit{multiple} \text{ sentences)} \\ 0 & \text{otherwise} \end{cases} \tag{7.6}$$

3) NEOF

$$\text{Sim}\ (R(i)_j, R(i)_k)\ (\text{NEOF}) = \begin{cases} 1 & \text{if the order of named entity categories of a} \\ & \text{relation is equal to that of the other relation} \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

4) NEVPF

$$\text{Sim}\ (R(i)_j, R(i)_k)\ (\text{NEVPF}) = \begin{cases} 1 & \text{if the named entity number of both sides} \\ & \text{of a verb in a relation-related sentence is} \\ & \text{the same as that in another relation-related} \\ & \text{sentence separately} \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

Note that if there is more than one verb in a relation-related sentence, *Sim* $(R(i)_i, R(i)_k)$ *(NEVPF)* is equal to the average similarity value of all verbs.

5) NECF

$$\text{Sim } (R(i)_j, R(i)_k\,)\ (\text{NECF}) = \begin{cases} 1 & \text{if all contexts of named entities for two} \\ & \text{relations are the same} \\ 0.75 & \text{if only a preceding or following context} \\ & \text{is not the same} \\ 0.5 & \text{if two preceding and/or following contexts} \\ & \text{are not the same} \\ 0.25 & \text{if three preceding and/or following} \\ & \text{contexts are not the same} \\ 0 & \text{if all contexts of named entities for two} \\ & \text{relations are not the same} \end{cases} \tag{7.9}$$

6) VSPF

$$\text{Sim } (R(i)_j, R(i)_k\,)\ (\text{VSPF}) = \begin{cases} 1 & \text{if the distribution of verb(s) for two relations} \\ & \text{is identical (either within } \textit{one} \text{ or in } \textit{multiple} \\ & \text{sentences)} \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

7) NEEWPOF

$$\text{Sim } (R(i)_j, R(i)_k\,)\ (\text{NEEWPOF}) = \begin{cases} 1 & \text{if the sequence of POS of particles for} \\ & \text{two relations is the same. At the same} \\ & \text{time, the relative position between named} \\ & \text{entity and particle is also identical} \\ 0.75 & \text{if the sequence of that for two relations} \\ & \text{is the same, but only the part of relative} \\ & \text{position between them is identical} \\ 0.5 & \text{if only the sequence of that for two} \\ & \text{relations is the same} \\ 0.25 & \text{if only the part of relative position} \\ & \text{between them is identical} \\ 0 & \text{otherwise} \end{cases} \tag{7.11}$$

8) NEPF

Sim $(R(i)_j, R(i)_k)$ (NEPF) = ((the same POS length of the first named entities of two relations ÷ the shorter POS length of the first named entities) + (the same POS length of the second named entities of two relations ÷ the shorter POS length of the second named entities)) ÷ 2

(7.12)

Note that the same POS length means the length of the same POS tags from the beginning of named entities.

9) NECPF

$$\text{Sim}\,(R(i)_j, R(i)_k)\,(\text{NECPF}) = \begin{cases} 1 & \text{if all the POS tags of the context of corresponding named entities for two relations are the same} \\ 0.75 & \text{if a POS tag of the preceding or following context is different} \\ 0.5 & \text{if two POS tags of the preceding or following context are different} \\ 0.25 & \text{if three POS tags of the preceding or following context are different} \\ 0 & \text{if all the POS tags of the context of corresponding named entities for two relations are different} \end{cases}$$

(7.13)

10) SPF

Sim $(R(i)_j, R(i)_k)$ (SPF) = minimal (1, the sum of the lengths of same POS tag subsequences for two relation-related sentences ÷ 10)

(7.14)

11) VVF

$$\text{Sim}\ (R(i)_j, R(i)_k\,)\ (\text{VVF}) = \begin{array}{l}\text{maximal similarity value of valence of all verbs in} \\ \text{two relation-related sentences, which means that} \\ \text{their valence expressions have the same number} \\ \text{and semantic constraints as do obligatory arguments}\end{array} \qquad (7.15)$$

12) NECTF

$$\text{Sim}\ (R(i)j, R(i)k\ )\ (\text{NECTF}) = \begin{cases} & \text{if the concepts of two named entities are} \\ & \text{not combining (don't join with “+”):} \\ 1 & \text{the first and second primitives are the same} \\ 0.8 & \text{only the first primitive is the same} \\ 0 & \text{the first and second primitives are all not} \\ & \text{the same} \\ & \\ & \text{if all the concepts of two named entities} \\ & \text{are combining or one is combining, the} \\ & \text{other is not combining:} \\ & \text{maximal similarity value among the pairs} \\ & \text{of combined sub-concept depending on the} \\ & \text{above non-combining definition} \end{cases} \qquad (7.16)$$

13) VCTF

$$
\mathrm{Sim}\,(R(i)_j, R(i)_k)\,(VCTF) = \begin{cases} \text{If the first primitives of the concept of two verbs} \\ \text{are the same:} \\ 0.3 * \dfrac{\alpha}{d+\alpha} + 0.5 * \ \mathrm{Sim}(V_s(p_1), V_t(p_1)) + \\ 0.2 * \mathrm{Sim}(V_s(p_2), V_t(p_2)) \\ \text{If those are \textit{not} the same:} \\ 0.8 * \dfrac{\alpha}{d+\alpha} + 0.2 * \mathrm{Sim}(V_s(p_2), V_t(p_2)) \\ \text{If there is \textit{no} second primitive of the concept, for} \\ \text{the first case above:} \\ 0.5 * \dfrac{\alpha}{d+\alpha} + 0.5 * \mathrm{Sim}(V_s(p_1), V_t(p_1)) \\ \text{for the second case above:} \\ \dfrac{\alpha}{d+\alpha} \end{cases} \tag{7.17}
$$

$$
\mathrm{Sim}(V_s(p_v), V_t(p_v)) = \begin{cases} 1 & \text{if two primitives of concept are the same} \\ 0.5 & \text{if \textit{only} the part of two primitives are the same} \\ 0 & \text{if two primitives of concept are \textit{absolutely not} the same} \end{cases} \tag{7.18}
$$

Here, V_s and V_t are two relation-related verbs. $1 \leq s \leq tnv_1$; $1 \leq t \leq tnv_2$. tnv_1 and tnv_2 are two total numbers of verbs corresponding to two matched relations respectively. $1 \leq v \leq 2$. p_1 and p_2 are the first and second primitive of verb concepts respectively. α is an adjust constant, we set it as 1.6 depending on (Liu and Li, 2002). d is the semantic distance of two primitives of verb concepts in Sports Ontology. Finally, the maximal concept similarity value among verb pairs related two relations is chosen as the value of *Sim (R(i)$_j$, R(i)$_k$) (VCTF)*.

In the calculation formulas (7.5) - (7.17), $1 \le i \le 14$; $1 \le j, k \le m, j \ne k$. m is the total number of the relation *R(i)* in the NER pattern library.

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

In the next section, we will give other related definitions with respect to PNCBL and the learning algorithm.

7.3.4 Learning Algorithm

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

Definition 7.8 (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 7.9 (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 7.10 (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.

For a relation, a feature weight is the *proportion* of the average similarity of this feature to the sum of the average similarities of all features, that is,

$$\mathrm{f(s)_w(R(i))} = \frac{\mathrm{Sim_{average}f(s)(R(i))}}{\sum_{t=1}^{n} \mathrm{Sim_{average}f(t)(R(i))}} \tag{7.19}$$

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

$$\mathrm{Sim_{average}f(s)(R(i))} = \frac{\sum_{1 \le j,\, k \le m;\, j \ne k} \mathrm{Sim\ (R(i)_j, R(i)_k)\ (f(s))}}{\mathrm{Sum_{relation\_pair}(R(i)_j, R(i)_k)}} \tag{7.20}$$

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 (7.4).

Definition 7.11 (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. In other words, the similarity value between them must be greater than or equal to the threshold.

It is calculated by the *average* of the *sum* of average similarity values for *selected* features. The calculation formula is shown in formula (7.21).

$$\text{IdenThrh}(R(i)) = \frac{\sum_{t=1}^{n} \text{Sim}_{\text{average}} f(t)(R(i))}{n} \tag{7.21}$$

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

In the previous three subsections, the relation features, NER and non-NER patterns, and the similarity calculation for NERs have been elaborated. Moreover, a number of basic conceptions are defined in this subsection. Based on this preliminary knowledge, the PNCBL algorithm is summarized as follows:

PNCBL Algorithm:

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.

During the building of NER and non-NER patterns, all sentence groups in the texts, *except for* those in which there exist fewer than two named entities, are accepted as either an NER or a non-NER pattern, or both.

For the selection of features, there are different criteria for NERs and non-NERs. We suggest that the GCF sets serve as the NER identification and the ICF sets play a role in the non-NER identification, because the effect of the GCFs is to capture as many *correct* NERs as possible, and the task for the ICFs is to remove non-NERs as *accurately* as possible.

After that, according to Definition 7.10, we can calculate weights for different features. Moreover, based on Definition 7.11, we can determine identification thresholds for each NER and non-NER.

In Appendix E, an example demonstrating how to compute the feature similarity and the self-similarity of a NER, select features, calculate corresponding feature weights and determine the identification threshold for this NER is offered.

In addition, the NER and non-NER patterns are stored in two hash tables, the index for a NER or non-NER includes one number of a *sentence group* and two numbers of *named entities* in the sentence group. Its structure is shown in Figure 7.6.

PS_TM	no. of s. g.	ne_no.1	ne_no.2	no. of s. g.	ne_no.1	ne_no.2	…
PS_CP	no. of s. g.	ne_no.1	ne_no.2	no. of s. g.	ne_no.1	ne_no.2	…
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
LOC_CPC	no. of s. g.	ne_no.1	ne_no.2	no. of s. g.	ne_no.1	ne_no.2	…

Figure 7.6 Relation and Non-Relation Index Structure

So far, the preconditions, procedure and outcome of the learning have been described. In the next section, an approach for NER identification, which directly utilizes the outcome of the learning, is proposed.

7.4 Named Entity Relation Identification

As we have described in the last section, in PNCBL, positive and negative cases have been transformed into NER and non-NER patterns. Both patterns indicate all information of NER and non-NER features. According to the self-similarity of an NER and non-NER, GCF and ICF sets are separately determined. Furthermore, the weights of GCFs and ICFs as well as the identification thresholds of NERs and non-NERs are decided as well.

The approach of the NER identification is based on PNCBL, that is to say, it can utilize the outcome of the learning for further identifying NERs and removing non-NERs. But it is also confronted with some problems, such as, *NER identification tradeoff*, *NER conflicts*, and *NER omissions*. In a sense, the procedure of the NER identification is a procedure solving the above problems, which is illustrated in Figure 7.7.

In Subsection 7.4.1, an *optimal* tradeoff for NER identification is proposed. It takes account of two aspects of the identification performance, i.e., *recall* and *precision*. After that, Subsection 7.4.2 describes the solutions for NER conflicts. To infer missing NERs in identification results, Subsection 7.4.3 presents the inference precondition and issue. Finally, the NER identification algorithm will be summarized in Subsection 7.4.4.

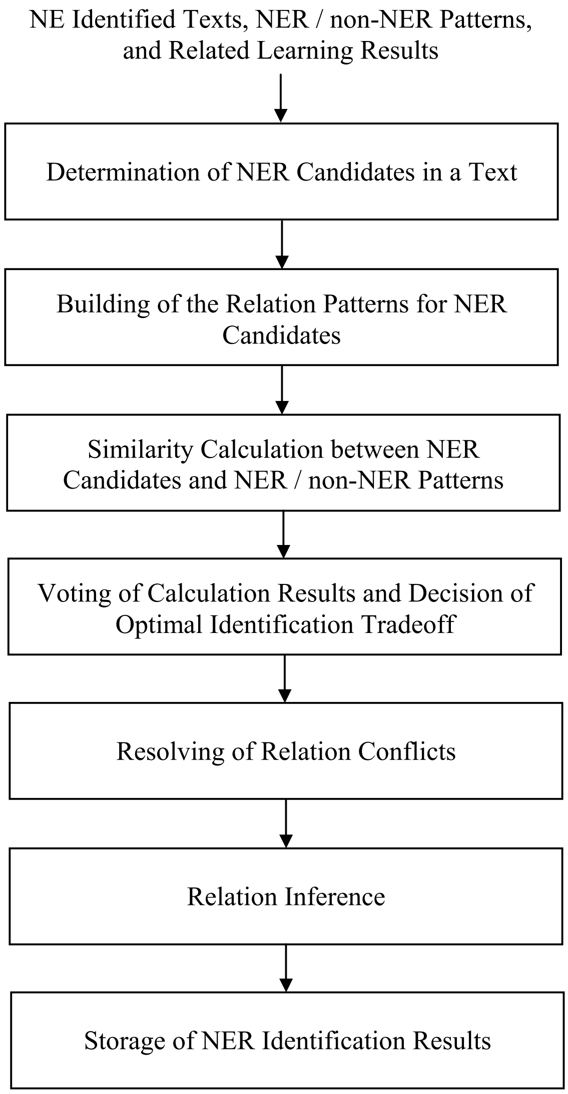

NE Identified Texts, NER / non-NER Patterns, and Related Learning Results
Determination of NER Candidates in a Text
Building of the Relation Patterns for NER Candidates
Similarity Calculation between NER Candidates and NER / non-NER Patterns
Voting of Calculation Results and Decision of Optimal Identification Tradeoff
Resolving of Relation Conflicts
Relation Inference
Storage of NER Identification Results

Figure 7.7 Identification Procedure of NERs

7.4.1 Achieving an 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-NER*s* in the non-NER pattern library. The computing formulas in this procedure are listed as follows:

$$\mathrm{Sim}\ (R(i)_{can}, R(i)_{j1}) = \sum_{k1=1}^{\mathrm{Sum(GCF)}_i} \{\ w_i\,(GCF_{k1}) * \mathrm{Sim}\ (R(i)_{can}, R(i)_{j1})\ (GCF_{k1})\ \} \qquad (7.22)$$

and

$$\mathrm{Sim}\ (R(i)_{can}, NR(i)_{j2}) = \sum_{k2=1}^{\mathrm{Sum(ICF)}_i} \{\ w_i\,(ICF_{k2}) * \mathrm{Sim}\ (R(i)_{can}, NR(i)_{j2})\ (ICF_{k2})\ \} \qquad (7.23)$$

Where *R(i)* represents the NER$_i$, and *NR(i)* expresses the non-NER$_i$, $1 \leq i \leq 14$. $R(i)_{can}$ 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. $1 \leq j1 \leq Sum\ (R(i))$ and $1 \leq j2 \leq Sum\ (NR(i))$. *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. From this, we can see that this tradeoff process seems to play at seesaw. Therefore, balancing the matching results between NERs / non-NERs and a NER candidate is one of critical tasks in the identification procedure.

For the sake of further explanation of how to make the optimal tradeoff, we will illustrate the *voting conditions* in detail. First, a number of the conceptions used in the conditions are defined in the following:

Definition 7.12 (TotalFS):

$TotalFS_1(R(i))$ and $TotalFS_2$(NR(i)) are two sums of the overall similarity results (see formulas (7.22) and (7.23)) between a NER candidate and NER / non-NER patterns separately, whose overall similarity results must be greater than the identification threshold of NER and non-NER respectively and where *R(i)* and *NR(i)* ($1 \leq i \leq 14$) are a predefined NER and non-NER respectively.

Definition 7.13 (TotalFSNum):

$TotalFSNum_1(R(i))$ and *$TotalFSNum_2(NR(i))$* are two counters to record the number of the matched pairs between a NER candidate and NER / non-NER patterns separately, whose overall similarity results must meet the condition in Definition 7.12.

Definition 7.14 (FS065Num):

$FS065Num_1(R(i))$ and *$FS065Num_2(NR(i))$* count the number of the matched pairs between a NER candidate and NER / non-NER patterns respectively, whose overall similarity results must be greater than 0.65.

Definition 7.15 (FS075Num):

$FS075Num_2$ records the number of the matched pairs between an NER candidate and non-NER patterns, whose overall similarity results must be greater than 0.75.

Definition 7.16 (AveTotalFS):

$AveTotalFS_1(R(i))$ and *$AveTotalFS_2(R(i))$* are two average values of $TotalFS_1$(R(i)) and $TotalFS_2$(NR(i)) respectively, that is:

$$\mathrm{AveTotalFS}_1(\mathrm{R(i)}) = \mathrm{TotalFS}_1(\mathrm{R(i)}) \div \mathrm{TotalFSNum}_1(\mathrm{R(i)})$$

$$\mathrm{AveTotalFS}_2(\mathrm{NR(i)}) = 2 \times \mathrm{TotalFS}_2(\mathrm{NR(i)}) \div \mathrm{TotalFSNum}_2(\mathrm{NR(i)})$$

Note that since the selected features for non-NERs are ICFs, their average similarity values are *lower* than the self-similarity of non-NERs. Therefore, the result of $\mathrm{TotalFS}_2(\mathrm{NR(i)}) \div \mathrm{TotalFSNum}_2(\mathrm{NR(i)})$ should be enlarged, e.g., double the result (according to our experiments), so that $\mathrm{AveTotalFS}_2(\mathrm{NR(i)})$ is *comparable* with $\mathrm{AveTotalFS}_1(\mathrm{R(i)})$.

Second, the *voting judgment* consists of the following combined conditions which are derived from experimental observations. Once the conditions are met, a NER candidate is regarded as a real NER.

```
$\mathrm{FS075Num}_2(\mathrm{NR(i)}) = 0$
and
{   $\mathrm{TotalFSNum}_1(\mathrm{R(i)}) > \mathrm{TotalFSNum}_2(\mathrm{NR(i)})$
    or
    {   $\mathrm{R(i)} \neq \mathrm{HT\_VT}$
        and
        $\mathrm{R(i)} \neq \mathrm{WT\_LT}$
        and
        $\mathrm{R(i)} \neq \mathrm{DT\_DT}$
    }
    and
    {   $\mathrm{Abs}(\mathrm{TotalFSNum}_1(\mathrm{R(i)}) - \mathrm{TotalFSNum}_2(\mathrm{NR(i)})) \leq 9$
        and
        {   $\mathrm{TotalFSNum}_2(\mathrm{NR(i)}) \neq 0$
            and
            $\mathrm{Abs}(\mathrm{FS065Num}_1(\mathrm{R(i)}) - \mathrm{FS065Num}_2(\mathrm{NR(i)})) < 5$
            and
            $\mathrm{FS065Num}_1(\mathrm{R(i)}) > \mathrm{FS065Num}_2(\mathrm{NR(i)})$
            and
            $\mathrm{TotalFSNum}_1(\mathrm{R(i)}) \neq 0$
            and
            $\mathrm{TotalFSNum}_2(\mathrm{NR(i)}) \neq 0$
            and
            $\mathrm{AveTotalFS}_1(\mathrm{R(i)}) > \mathrm{AveTotalFS}_2(\mathrm{NR(i)})$
```

```
                or
                $FS065Num_1(R(i)) > FS065Num_2(NR(i))$
                and
                $Abs(FS065Num_1(R(i)) - FS065Num_2(NR(i))) \geq 5$
            }
        }
        or
        {   $R(i) = HT\_VT$
            or
            $R(i) = WT\_LT$
            or
            $R(i) = DT\_DT$
        }
        and
        {   $Abs(FS065Num_1(R(i)) - FS065Num_2(NR(i))) < 5$
            and
            $FS065Num_1(R(i)) > FS065Num_2(NR(i))$
            and
            $TotalFSNum_1(R(i)) \neq 0$
            and
            $TotalFSNum_2(NR(i)) \neq 0$
            and
            $AveTotalFS_1(R(i)) > AveTotalFS_2(NR(i))$
        }
        or
        {   $FS065Num_1(R(i)) > FS065Num_2(NR(i))$
            and
            $Abs(FS065Num_1(R(i)) - FS065Num_2(NR(i))) \geq 5$
        }
    }
```

In the above combined conditions, Abs() is a function of the absolute value. The thresholds in the conditions, such as 0.65, 0.75, 5, and 9, depend on (Cover and Hart, 1967; Duda and Hart, 1973) and our experiments.

The voting in our approach refers to the similarity calculation results between an NER candidate and NER / non-NER patterns. It pays special attention to circumstance in which both similarity calculation results between a NER candidate and NER / non-NER patterns are very close. If this happens, it exploits *multiple* calculation results to measure and arrive at a final decision. For instance, although $\text{FS065Num}_1(\text{R(i)})$ is greater than $\text{FS065Num}_2(\text{NR(i)})$ (R(i) = HT_VT, WT_LT, or DT_DT), if the difference between $\text{FS065Num}_1(\text{R(i)})$ and $\text{FS065Num}_2(\text{NR(i)})$ is less than five, which means that both results are close, the difference between $\text{AveTotalFS}_1(\text{R(i)})$ and $\text{AveTotalFS}_2(\text{NR(i)})$ will be observed further. Additionally, notice that the impact of non-NER patterns is to restrict possible misidentified non-NERs. For example, if an NER candidate's similarity result matched with a non-NER pattern is equal to or more than 0.75. Obviously, this NER candidate is very similar to the non-NER. In this situation, the NER candidate is *forbidden* to pass by means of one of the conditions.

On the other hand, the voting assigns different thresholds (e.g., 5 or 9) to different NER candidates (HT_VT, WT_LT, and DT_DT or other NERs). Because the former three NERs have the same kind of NEs, that is, they all have two TNs, the identification for these NERs is more difficult than for others. Thus, when voting, the corresponding threshold should be set more *strictly*.

After voting, however, there are still some special problems remaining in the identification results, such as relation conflicts and relation omissions. The next two subsections will briefly describe how to settle NER conflicts and infer missing NERs respectively.

7.4.2 Resolving Relation Conflicts

Generally speaking, the voting procedure is a decision-making procedure of accepting or refusing NER candidates. 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. For example, the relation conflict is one of the problems, which means that *contradictory* NERs occur in identification results. For the NERs defined in Subsection 7.2.1, there are two types of NER conflicts:

(i) *Same Relations*

The same kind of relations with different argument position:

e.g., the relations HT_VT, WT_LT and LOC_CPC, that is,

HT_VT(*ne1*, *no1*; *ne2*, *no2*) and *HT_VT*(*ne2*, *no2*; *ne1*, *no1*); or

WT_LT(*ne1*, *no1*; *ne2*, *no2*) and *WT_LT*(*ne2*, *no2*; *ne1*, *no1*); or

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

(ii) *Different Relations*

The different kinds of relations with same or different argument positions:

e.g., the relations WT_LT and DT_DT, i.e.,

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 named entities 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 relations, which has lower average similarity value with NER patterns or higher average similarity value with non-NER patterns. As an example, the conflict of the relation HT_VT is resolved by the following conditions, which are expressed by pseudo code:

```
if (AveTotalFS$_{11}$(HT_VT) > AveTotalFS$_{21}$(HT_VT) and
   FS065Num$_{11}$ (HT_VT) ≥ FS065Num$_{21}$(HT_VT) or
   AveTotalFS$_{22}$(non-HT_VT) > AveTotalFS$_{12}$(non-HT_VT) and
   FS065Num$_{22}$(non-HT_VT) ≥ FS065Num$_{12}$(non-HT_VT))
   remove HT_VT(ne2, no2; ne1, no1);

if (AveTotalFS$_{21}$(HT_VT) > AveTotalFS$_{11}$(HT_VT) and
   FS065Num$_{21}$(HT_VT) ≥ FS065Num$_{11}$(HT_VT) or
   AveTotalFS$_{12}$(non-HT_VT) > AveTotalFS$_{22}$(non-HT_VT) and
   FS065Num$_{12}$(non-HT_VT) ≥ FS065Num$_{22}$(non-HT_VT))
   remove HT_VT(ne1, no1; ne2, no2);
```

In the above conditions, FS065Num$_{11}$(HT_VT), FS065Num$_{12}$(non-HT_VT), AveTotalFS$_{11}$(HT_VT), and AveTotalFS$_{12}$(non-HT_VT) correspond to the relation *HT_VT(ne1, no1; ne2, no2)*. Analogously, FS065Num$_{12}$(non-HT_VT), FS065Num$_{22}$(non-HT_VT), AveTotalFS$_{21}$(HT_VT), and AveTotalFS$_{22}$(non-HT_VT) have a correspondence to the relation *HT_VT(ne2, no2; ne1, no1)*. If the first combined conditions are met, the relation HT_VT(ne2, no2; ne1, no1) is removed; Otherwise, the relation HT_VT(ne1, no1; ne2, no2) is deleted.

7.4.3 Inferring Missing Relations

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

Note that sometimes the relation inference is *irreversible*. The relations from the above example can prove this conclusion. If identified relations include

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

Obviously, we *cannot* infer the tenability of the following NER:

PS_ID (ne3, no3; ne1, no1)

This is due to the fact that if there is a position in a sports team and a person is a member of this sports team, we cannot say that this person certainly occupies this position.

So far, we have introduced principal strategies used in NER identification. As a whole description, the algorithm of the relation identification will be depicted in the next subsection.

7.4.4 Algorithm of Relation Identification

The NER identification algorithm can be summarized in the following:

1) Input NE identified texts, NER and non-NER patterns (store the patterns in hash tables and construct corresponding indexes for them), and related learning results;
2) Detect whether there exist NER candidates in a text;
3) Record the positions of NEs and possible types of NER candidates;
4) Build the relation patterns for NER candidates;
5) Store the above patterns in a hash table and construct the indexes for them;
6) Match these NER candidates using NER, non-NER patterns, and Sports Ontology, as well as feature weights and identification thresholds;
7) Decide optimal identification tradeoff using the voting mode;
8) Resolve NER conflicts;
9) Based on existing NERs infer missing NERs;
10) Store final identification results.

It is necessary to explicate some key points in the algorithm:

(i) To match NER candidates *conveniently*, we also transform them into NER patterns;

(ii) For the sake of *speeding up* the retrieval for NER candidate patterns, we adopt *hash tables* for storing those and then construct indexes for them. The index structure is the same as that introduced in Subsection 7.3.4;

(iii) Sports Ontology provides the concepts of NEs and verbs (see Chapter 5). Moreover, we can compute semantic distance of the concepts for two verbs by it;

(iv) In the sequence arranged from the seventh to ninth step, the performance of NER identification can be improved further.

7.5 Experimental Results and Evaluation

The experiments we finally accomplished are associated with NER identification, which contain learning and identification, such as building NER and non-NER pattern libraries, computing NER and non-NER self-similarity, selecting GCFs and ICFs for different NERs and non-NERs, calculating feature weights, determining identification thresholds, and identifying NERs. Note that the learned texts were taken from the Jie Fang Daily in *2001*; while the tested texts were *randomly* chosen from the Jie Fang Daily in *2002*.

In order to explain how to measure the performance for NER identification, below, in Subsection 7.5.1 and 7.5.2, we elaborate on related experiment conditions and results step by step with data presentation.

7.5.1 Experimental Conditions

The main resources are used for learning and identification are NER and non-NER patterns. Before learning, more than 50 texts from the Jie Fang Daily in 2001 were annotated based on the NE identification. As described in Section 5.6, here we also adopted double-person annotation method for the NER annotation. During learning, both pattern libraries are established in terms of the annotated texts and Sports Ontology. They have *142* (*534 NERs*) and *98* (*572 non-NERs*) *sentence groups* respectively. The NER and non-NER *distributions* in the corresponding pattern library are shown in Table 7.5 and Table 7.6. In these two tables we can see that the *occurrence frequency* of some NERs or non-NERs is remarkably different. In the former library, five NERs exceed the *average relation number* (38.14), but another nine are lower than that; while in the latter library, six non-NERs override the average relation number (40.86), eight NERs are *not* over that. In fact, it is a *normal* natural language phenomenon and reflects the expression practice of journalists in news.

TM_CP	**PS_ID**	**ID_TM**	**PS_TM**	**WT_LT**	**CP_TI**	**CP_LOC**
102	72	66	66	41	30	26
HT_VT	**PS_CP**	**TM_CPC**	**PS_CPC**	**DT_DT**	**CP_DA**	**LOC_CPC**
26	26	23	21	13	12	10

Table 7.5 Relation Distribution in Relation Pattern Library

HT_VT	**WT_LT**	**PS_ID**	**TM_CPC**	**ID_TM**	**DT_DT**	**LOC_CPC**
111	112	65	54	45	42	40
PS_TM	**CP_LOC**	**CP_TI**	**PS_CPC**	**CP_DA**	**PS_CP**	**TM_CP**
27	18	18	17	10	7	6

Table 7.6 Non-Relation Distribution in Non-Relation Pattern Library

7.5.2 Experimental Results

The feature selection is to pick up the features used for providing the most important information for the identification of NERs or non-NERs. According to the learning algorithm in Subsection 7.3.4, NERs and non-NERs have corresponding GCF and ICF sets, which are determined by the self-similarity of NERs and the average similarity of features (see formulas (7.2) - (7.4) in Subsection 7.3.3). Table 7.7 and 7.8 show the GCF and ICF sets for NERs and non-NERs separately. In these two tables, we can observe that some features are both GCF and ICF. In addition, in general, the *size* of GCF sets is larger than that of ICF sets.

PS_TM	PS_CP	PS_CPC	PS_ID	HT_VT	WT_LT	DT_DT
f_1 f_2 f_3 f_6 f_8 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{11} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{12}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{11} f_{12} f_{13}
TM_CP	**TM_CPC**	**ID_TM**	**CP_DA**	**CP_TI**	**CP_LOC**	**LOC_CPC**
f_1 f_2 f_3 f_6 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{12}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{12}	f_1 f_2 f_3 f_6 f_8 f_{10} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_{10} f_{11} f_{12} f_{13}	f_1 f_2 f_3 f_6 f_8 f_{12} f_{13}

Table 7.7 Selected Features (GCFs) for 14 Relations

PS_TM	PS_CP	PS_CPC	PS_ID	HT_VT	WT_LT	DT_DT
f_4 f_5 f_7 f_9 f_{11}	f_2 f_4 f_5 f_7 f_9 f_{10} f_{11}	f_4 f_5 f_7 f_9 f_{11}	f_4 f_5 f_7 f_9 f_{11}	f_4 f_5 f_7 f_9 f_{11}	f_4 f_5 f_7 f_9 f_{11}	f_4 f_5 f_7 f_9 f_{11}
TM_CP	**TM_CPC**	**ID_TM**	**CP_DA**	**CP_TI**	**CP_LOC**	**LOC_CPC**
f_4 f_5 f_6 f_7 f_8 f_9	f_4 f_5 f_7 f_8 f_9 f_{11}	f_4 f_5 f_7 f_9 f_{11}	f_4 f_5 f_7 f_9 f_{11}	f_3 f_4 f_5 f_7 f_8 f_9	f_4 f_5 f_7 f_8 f_9	f_4 f_5 f_7 f_9 f_{11}

Table 7.8 Selected Features (ICFs) for 14 Non-Relations

Note: f_1 = STF; f_2 = NESPF; f_3 = NEOF; f_4 = NEVPF; f_5 = NECF; f_6 = VSPF; f_7 = NEPPOF; f_8 = NEPF; f_9 = NECPF; f_{10} = SPF; f_{11} = VVF; f_{12} = NECT; f_{13} = VCTF

After the above feature sets are determined, we can further calculate relevant feature weights (see formulas (7.19) and (7.20)). The feature weights corresponding to the GCFs and ICFs are listed in Table 7.9 and 7.10 respectively. For each NER or non-NER, the sum of all of the feature weights is consistently equal to 1. Generally speaking, the *bigger* a feature weight‘s value is, the *more important* the feature is for the identification of NER or non-NER.

	PS_TM	PS_CP	PS_CPC	PS_ID	HT_VT	WT_LT	DT_DT
f_{1w}	0.17343546	0.18417682	0.09008357	0.17474796	0.13937688	0.11321937	0.09624822
f_{2w}	0.11546655	0.11787316	0.09008357	0.13981642	0.12055806	0.11832701	0.09624822
f_{3w}	0.10717181	0.09638587	0.13257581	0.13142201	0.09409410	0.11832701	0.17874669
f_{4w}							
f_{5w}							
f_{6w}	0.11546655	0.11112002	0.10198140	0.18449630	0.13937688	0.12981921	0.12833096
f_{7w}							
f_{8w}	0.10267885	0.09561846	0.12365244	0.19406410	0.11173675	0.09649543	0.09223788
f_{9w}							
f_{10w}	0.10045142	0.11234780	0.10053666		0.09033028	0.09698138	
f_{11w}			0.08498450				0.11458122
f_{12w}	0.18010930	0.17324899	0.16724962	0.17545319	0.18736486	0.17072289	0.17003852
f_{13w}	0.10522009	0.10922898	0.10885239		0.11716224	0.15610766	0.12356834
	TM_CP	**TM_CPC**	**ID_TM**	**CP_DA**	**CP_TI**	**CP_LOC**	**LOC_CPC**
f_{1w}	0.14870416	0.13033457	0.18347110	0.13783681	0.16985454	0.11625134	0.18277073
f_{2w}	0.11008418	0.10961916	0.09989607	0.11548490	0.09640393	0.11047088	0.12184716
f_{3w}	0.11146347	0.13033457	0.12944280	0.16763937	0.11142791	0.10597497	0.12184716
f_{4w}							
f_{5w}							
f_{6w}	0.13685092	0.12170315	0.12615982	0.13783681	0.13980657	0.14129996	0.12184716
f_{7w}							
f_{8w}			0.14813462	0.12231465	0.09527914		0.10407778
f_{9w}							
f_{10w}	0.13462590	0.15199943	0.11657377	0.11399480	0.09506843	0.11143427	
f_{11w}						0.09056043	
f_{12w}	0.21973713	0.21751201	0.19632180	0.20489256	0.19355902	0.20655486	0.22237103
f_{13w}	0.13853423	0.13849711			0.09860057	0.11745327	0.12523898

Table 7.9 Feature Weights for 14 Relations

	PS_TM	PS_CP	PS_CPC	PS_ID	HT_VT	WT_LT	DT_DT
f_{1w}							
f_{2w}		0.20524517					
f_{3w}							
f_{4w}	0.08791222	0.06841505	0.14503220	0.10443031	0.08846026	0.08394295	0.07563738
f_{5w}	0.07587694	0.06271380	0.06246341	0.05688181	0.06346348	0.06441595	0.09917720
f_{6w}							
f_{7w}	0.20729268	0.15393387	0.26156550	0.20342262	0.31517762	0.30425120	0.33472310
f_{8w}							
f_{9w}	0.21276832	0.14253137	0.24204569	0.18582241	0.15571652	0.16011688	0.22392353
f_{10w}		0.20752566					
f_{11w}	0.41614980	0.15963513	0.28889322	0.44944283	0.37718216	0.38727298	0.26653874
f_{12w}							
f_{13w}							

	TM_CP	TM_CPC	ID_TM	CP_DA	CP_TI	CP_LOC	LOC_CPC
f_{1w}							
f_{2w}							
f_{3w}					0.34176600		
f_{4w}	0.11538462	0.07211530	0.06160856	0.06469997	0.05858288	0.08413938	0.11173789
f_{5w}	0.07692309	0.03149784	0.05156675	0.06537044	0.04330597	0.06642582	0.04918580
f_{6w}	0.23076925						
f_{7w}	0.17307694	0.19237110	0.28469790	0.26651025	0.12406575	0.23598646	0.27552380
f_{8w}	0.24038462	0.32213690			0.32694050	0.39494234	
f_{9w}	0.16346155	0.13562292	0.17353602	0.20113981	0.10533884	0.21850598	0.28511090
f_{10w}							
f_{11w}		0.24625583	0.42859074	0.40227962			0.27844160
f_{12w}							
f_{13w}							

Table 7.10 Feature Weights for 14 Non-Relations

Similarly, the identification thresholds can be decided according to the formula (7.21). Table 7.11 and 7.12 give the identification thresholds for NERs and non-NERs respectively. As we mentioned previously, because the used features of non-NERs are ICFs whose average similarity is less than or equal to the self-similarity of corresponding NER, we can see that all of the identification thresholds of non-NERs in Table 7.12 are less than that of NERs in Table 7.11.

PS_TM	PS_CP	PS_CPC	PS_ID	HT_VT	WT_LT	DT_DT
0.61824787	0.62648827	0.62258476	0.72240570	0.65400980	0.71628696	0.69931364
TM_CP	**TM_CPC**	**ID_TM**	**CP_DA**	**CP_TI**	**CP_LOC**	**LOC_CPC**
0.64343750	0.65418226	0.71002865	0.69722950	0.64413120	0.59883520	0.62529550

Table 7.11 Identification Thresholds for 14 Relations

PS_TM	PS_CP	PS_CPC	PS_ID	HT_VT	WT_LT	DT_DT
0.18210623	0.2982993	0.18834558	0.18848124	0.16841121	0.16558047	0.14989770
TM_CP	**TM_CPC**	**ID_TM**	**CP_DA**	**CP_TI**	**CP_LOC**	**LOC_CPC**
0.28888887	0.22370915	0.16356166	0.22096296	0.23267585	0.18694991	0.15378633

Table 7.12 Identification Thresholds for 14 Non-Relations

To test the performance of our approach, we *randomly* choose 32 sentence groups from the *Jie Fang Daily* in *2002* (these sentence groups are *out of* either NER or non-NER pattern library), which embody 117 different NER candidates. The NER candidate *distribution* in these sentence groups is depicted in Table 7.13 below.

PS_ID	**CP_LOC**	**TM_CP**	**ID_TM**	**PS_TM**	**PS_CPC**	**CP_TI**
15	14	14	12	12	8	7
HT_VT	**TM_CPC**	**LOC_CPC**	**PS_CP**	**WT_LT**	**CP_DA**	**DT_DT**
7	6	5	5	5	3	3

Table 7.13 Relation Candidate Distribution

In the evaluation of the performance for the NER identification, the following formulas are utilized to compute recall and precision (F-measure's computing formula is the same as the formula (4.3)):

$$\text{Recall} = \frac{\text{correct number of identified NERs}}{\text{total number of NERs}} \qquad (7.24)$$

$$\text{Precision} = \frac{\text{correct number of identified NERs}}{\text{number of identified NERs}} \qquad (7.25)$$

where the NER is defined as the same kind of NERs.

For evaluating the effects of *negative cases*, we made two experiments to compare its impact on two sides: the first one is *only* to use *positive cases* to do learning and identification; the second one employs *both cases* to do that. The trade-off combined conditions of the former identification are described as follows. Compared with those of employing both cases in Subsection 7.4.1, we can observe that learning results with respect to negative cases are not applied to the following combined conditions. Similarly, the conditions are derived from experimental observation.

```
{   $R(i) \neq HT\_VT$
    and
    $R(i) \neq WT\_LT$
    and
    $R(i) \neq DT\_DT$
}
and
$\mathrm{TotalFSNum}_1(R(i)) \geq 9$
and
{   $\mathrm{FS065Num}_1(R(i)) < 5$
    and
    $\mathrm{TotalFSNum}_1(R(i)) \neq 0$
    and
```

```
          $TotalFS_1(R(i)) \div TotalFSNum_1(R(i)) > 0.75$
          or
          $FS065Num_1(R(i)) \geq 5$
     }
     or
     {    $R(i) = HT\_VT$
          or
          $R(i) = WT\_LT$
          or
          $R(i) = DT\_DT$
     }
     and
     {    $FS065Num_1(R(i)) < 5$
          and
          $TotalFSNum_1(R(i)) \neq 0$
          and
          $TotalFS_1(R(i)) \div TotalFSNum_1(R(i)) > 0.75$
          or
          $FS065Num_1(R(i)) \geq 5$
     }
```

Table 7.14 and 7.15 show the average and total average recall, precision, and F-measure for 14 different NERs *only* by *positive* case-based learning and identification respectively. Table 7.16 and 7.17 demonstrate the average and total average recall, precision, and F-measure for all NERs by *positive and negative* case-based learning and identification separately. 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 7.16 are *higher than* those of corresponding NERs in Table 7.14; 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 7.14 are lower than those of corresponding NERs in Table 7.16. 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.

In the final section, we will discuss primary differences between our proposed approach and other relation identification approaches and compare their identification performances of relations.

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

Table 7.14 Average Recall, Precision, and F-measure for 14 Relations (only by Positive Case-Based Learning and Identification)

Total Average Recall	71.99
Total Average Precision	56.98
Total Average F-measure[32]	63.61

Table 7.15 Total Average Recall, Precision and F-Measure for 14 Relations (only by Positive Case-Based Learning and Identification)

[32] Total Average F-measure = 2 × Total Average Recall × Total Average Precision ÷ (Total Average Recall + Total Average Precision)

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

Table 7.16 Average Recall, Precision and F-Measure for 14 Relations (by Positive and Negative Case-Based Learning and Identification)

Total Average Recall	78.50
Total Average Precision	63.92
Total Average F-measure	70.46

Table 7.17 Total Average Recall, Precision and F-Measurefor 14 Relations (by Positive and Negative Case-Based Learning and Identification)

7.6 Discussion

Daelemans et al. (2000) have utilized *Information Gain* (Shannon, 1948), *Gain Ratio* (Quinlan, 1993), *Chi-squared* (White and Liu, 1994) as different feature weighting methods in their TiMBL system, which is an implementation combining a number of different memory-based learning methods and using some optimized means to make them more efficient. But we note that these feature weighting methods are defined for each feature independently. Thus, the relationship (e.g., feature weight proportion in a feature set) among feature weights, which expresses the importance of various features, is not considered.

In our approach, the average similarity of a feature (for the same kind of NERs) is used to estimate the relevant degree between the feature and that kind of NERs. According to the computing result, therefore, the difference in features can be *explicitly* observed. On the other hand, based on the average similarity of features, the self-similarity of an NER reflects the comprehensive impact of all features on the relation. Thus, it becomes a *principal standard* for us to select features for the NER identification. As a result, the weights of selected features are computed depending on their *percentage* of the average similarity of features in the selected feature set, which embodies feature importance for the relation.

Zhang and Zhou (2000) have employed *memory-based learning* to extract Chinese named entities and their relations. In their relation extraction method, the relation classes include *employee-of*, *location-of*, *product-of* and *no-relation*, and the feature set cites the definition from Soderland (1996). They utilized *IG-Tree* (Daelemans et al., 2000) to classify and identify different relations. The main shortcomings of their work are i) obviously, in their experiment the kind of defined relations is *simple* and of *small-size*, although the authors claim that they can expand the set, if training data are available; ii) it lacks a *quantitative measure* for the relationship between a feature and a relation, hence, it cannot view the importance for features as a whole; iii) the features are *not automatically* selected. Consequently, if the relations have a distinct difference, it seems that the same feature set for all relations is *not appropriate*. Compared with their method, our approach has considered and solved these problems.

According to (Dake, 2003), there are other related *voting* methods like *majority voting*, *distance weighted class voting* (Wettschereck, 1994) and *inverse linear weighting* (Dudani, 1976; Zavrel, 1997). They are derivatives of the *k-nearest neighbor classification* (Cover and Hart, 1967; Devijver and Kittler, 1982; Aha et al., 1991). Although the early idea of our NER identification approach also stems from k-nearest neighbor classification, we pay especial attention to the study of an optimal NER identification tradeoff utilizing *general* and *individual characters* between NERs and non-NERs. Except for that, we also employed other strategies to improve NER identification performance, such as *resolving NER conflicts* and *inferring missing NERs*.

Summarily, our approach including learning and identification has the following advantages:

- The defined *negative cases* are used to improve the NER identification performance as compared to *only* using positive cases;
- The NER and non-NER patterns, which are the alternative forms of positive and negative cases, have *human-readability* and *machine-operability*;
- 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;
- An idea of "*fuzzy*" computation is carried out in the definitions of feature similarity. Doubtless, this calculation conforms to properties of language;
- *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;
- 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;
- The primary calculations in learning and identification deal with *similarity calculations*, which are *simple*. Additionally, a great number of self-similarity calculations can be completed by *off-line mode* before the NER identification. Thus, it can save the *time* of the NER identification;
- 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.

- This approach can be applied to *sentence groups* containing *multiple* sentences. Thus identified NERs are allowed to cross sentence boundaries.

Finally, we have to acknowledge that it is difficult to compare the performances 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 NER identification adopting different methods for a comparison with our approach.

Zhang and Zhou (2000) viewed the entire identification problem as a series of classification problems and employed *memory-based learning* (MBL) to resolve them. Their system can extract three named entity relations (*employee-of*, *product-of*, and *location-of*). The performances for relation identification of MBL&I and PNCBL&I are listed in Table 7.18. In the table, we select similar NERs in our domain to correspond to the three types of the relations.

Method	**Processed Language**	**Relation Type**	**Recall**	**Precision**	**F-measure**
Memory-Based Learning and Identification	Chinese	employee-of	75.60	92.30	83.12
		product-of	56.20	87.10	68.32
		location-of	67.20	75.60	71.15
Positive and Negative Case-Based Learning and Identification	Chinese	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 7.18 Performances for Relation Identification

(MBL&I vs. PNCBL&I)

Roth and Yih (2002) suggested a *probabilistic inference model* (PIM) for recognizing named entities and their relations simultaneously, in which the classifiers are trained independently from *local* information in the sentence. Then this information, along with constraints induced among entity types and relations, is used to perform *global* inference that accounts for the *mutual dependencies* among the entities. The performances for the relation identification of PIM and PNCBL&I are shown in Table 7.19. Although there is *no person-person* relation defined in our research domain, we can infer such relations through *PS_TM* and *PS_ID* relations, e.g., *TeamCoach_TeamMember* relation. Therefore, we select these two NERs to correspond to the relation "*kill*".

Method	**Processed Language**	**Relation Type**	**Recall**	**Precision**	**F-measure**
Probabilistic Inference Model	English	kill	49.80	85.40	62.20
		born-in	87.60	70.70	78
		all (kill & born-in)	47.20 & 68.40	86.80 & 87.50	60.70 & 76.60
Positive and Negative Case-Based Learning and Identification	Chinese	PS_TM PS_ID	80 72.22	65 81.67	71.72 76.65
		PS_CPC	100	68.75	81.48
		PS_CP & CP_LOC	60 & 88.89	75 & 66.67	66.67 & 76.19

Table 7.19 Performances for Relation Identification

(PIM vs. PNCBL&I)

Shen and Chen (2002) utilized *Supertagger* (Joshi, 1994) and *Lightweight Dependency Analyzer* (Srinivas, 1997) to do shallow parsing on the texts. Then features are abstracted from the shallow parsing result, which are viewed as *weak hypotheses*. A *Boosting classifier* (Schapire, 2000) is used to combine all hypotheses into a single one. Finally, the result of the classifier is converted into a relation list. In this method, *almost no extra* annotation work is required. The performances for the relation identification of Supertag & LDA & Boosting and PNCBL&I are illustrated in Table 7.20. In the table, we choose three NERs (*PS_TM*, *PS_CP*, *PS_ID*) from our domain to correspond to the similar relation “*employee-of*”.

Method	**Processed Language**	**Relation Type**	**Recall**	**Precision**	**F-measure**
Supertag & LDA & Boosting	English	employee-of	81	76	78.42
Positive and Negative Case-Based Learning and Identification	Chinese	PS_TM PS_CP PS_ID	80 60 72.22	65 75 81.67	71.72 66.67 76.65

Table 7.20 Performances for Relation Identification

(Supertag & LDA & Boosting vs. PNCBL&I)

Using *kernel-based machine learning* (KBML) methods to extract relations from natural language sources, it is put forward by Zelenko et al. (2002). In these methods, they introduce *kernels* defined over shallow parse representations of text and design efficient algorithms for computing the kernels. In addition, they also use the devised kernels in conjunction with *Support Vector Machine* (Cortes and Vapnik, 1995) and *Voted Perception* (Freund and Schapire, 1999) learning algorithms for extracting relations. Table 7.21 shows the performances for relation identification of KBML and PNCBL&I.

From these tables we can deduce that the identification performance of relations for PNCBL&I is roughly comparable to the MBL&I, PIM, and Supertag & LDA & Boosting approaches.

Method	Processed Language	Relation Type	Recall	Precision	F-measure
Naïve Bayes	English	person-affiliation	75.59	91.88	82.93
Winnow			80.87	88.42	84.46
Voted Perceptron (contig)			79.58	89.74	84.34
SVM (contig.)			79.78	89.90	84.52
Voted Perceptron (sparse)			81.62	90.05	85.61
SVM (sparse)			82.73	91.32	86.80
Naïve Bayes		organization-location	71.94	90.40	80.04
Winnow			75.14	85.02	79.71
Voted Perceptron (contig)			64.43	92.85	76.02
SVM (contig.)			71.43	92.03	80.39
Voted Perceptron (sparse)			71	91.90	80.05
SVM (sparse)			76.33	91.78	83.30
Positive and Negative Case-Based Learning and Identification	Chinese	PS_TM	80	65	71.72
		PS_CP	60	75	66.67
		PS_ID	72.22	81.67	76.65
		CP_LOC	88.89	66.67	76.19
		TM_CPC	37.50	31.25	34.09

Table 7.21 Performances for Relation Identification (KBML vs. PNCBL&I)

Chapter 8

System Implementation

8.1 Overview

In this chapter, I would like to go into details for the system implementation. Based on the theories and approaches of Computer Science and Computer Linguistics presented from Chapter 3 to Chapter 7, we have implemented a prototype system called *Chinese Named Entity and Relation Identification System*, i.e. *CHINERIS*. The architecture of CHINERIS system is shown in Figure 8.1. The core components within the system embody word segmentation and POS tagging processing, named entity identification, and named entity relation identification. The system is implemented in Java 2 (version 1.4.1) (Sun Microsystems, Inc., 2003) using Windows 2000.

During the implementation, object-oriented design and programming methods (Tello, 1989; Winbald et al., 1990) are thoroughly used in the system development. Depending on the requirement of different algorithms, we define the corresponding classes to accomplish different tasks, which are software blueprints that define the variables and the methods common to all objects of a certain kind. Additionally, we also integrate other application systems and resources, e.g., *Modern Chinese Word Segmentation and POS Tagging System* (Liu, 2000) and *HowNet* (Dong and Dong, 2003) into the system as an aid in avoiding repeated development. Additionally, we utilize *Protégé-2000* (version 1.9) (Stanford Medical Informatics, 2003), an ontology editor and knowledge acquisition system, as a development environment for the implementation of *Sports Ontology*.

In the next section, I will present the system architecture, the class definitions, as well as invoked relationships and executed sequence of methods in the classes for every core component. In Section 8.3, the advantages of the system implementation methods will be discussed.

8.2 System Implementation

8.2.1 System Architecture

Before the CHINERIS system was implemented, we first implemented Chinese Named Entity Recognition System (CHINERS) (Yao et al., 2003) which is a part of CHINERIS system. In the CHINERS system, there are two major components:

- *Word Segmentation and POS Tagging Processing*

  It is implemented by integrating Modern Chinese Word Segmentation and the POS Tagging System (baseline system) and developing an error repairer for word segmentation and POS tagging. As a post-processing unit for the

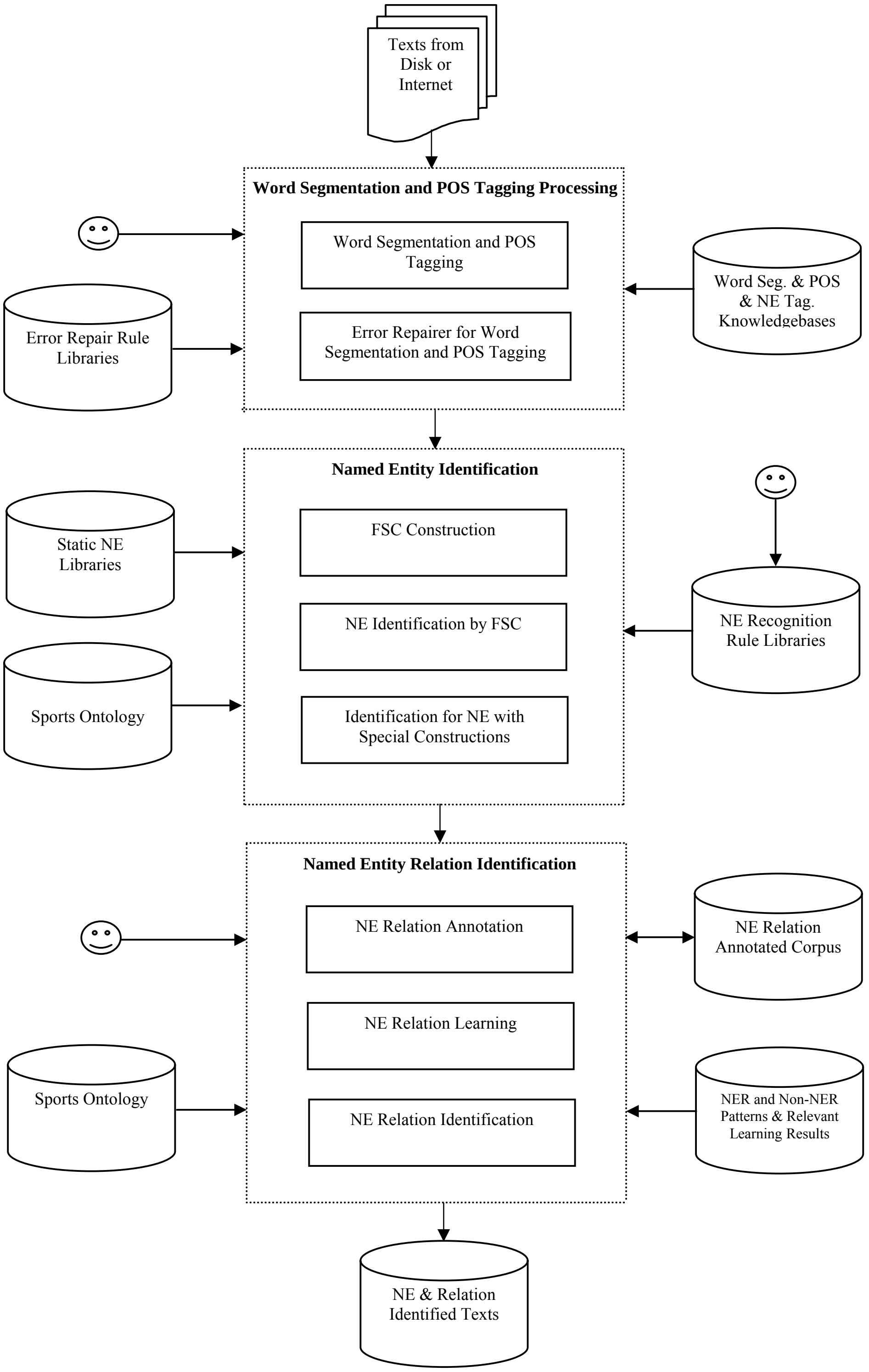

Texts from Disk or Internet
Word Segmentation and POS Tagging Processing
Word Segmentation and POS Tagging
Error Repairer for Word Segmentation and POS Tagging
Error Repair Rule Libraries
Word Seg. & POS & NE Tag. Knowledgebases
Named Entity Identification
FSC Construction
NE Identification by FSC
Identification for NE with Special Constructions
Static NE Libraries
Sports Ontology
NE Recognition Rule Libraries
Named Entity Relation Identification
NE Relation Annotation
NE Relation Learning
NE Relation Identification
Sports Ontology
NE Relation Annotated Corpus
NER and Non-NER Patterns & Relevant Learning Results
NE & Relation Identified Texts

Figure 8.1 Architecture of the CHINERIS System

baseline system, the error repairer rectifies the errors from word segmentation and the POS tagging procedure using a transformation-based error-driven machine learning method (see Chapter 4).

- *Named Entity Identification*

In order to effectively identify NEs, we implement a *shallow parser* that consists of FSC with three recognition levels[33] in this component, in which FSC is automatically constructed in terms of NE recognition rules. After that, TN, CT and PI can be identified using the FSC. Considering the fact that some NEs have special constructions, a unit called "Identification for NEs with Special Constructions" is developed for identifying TN and CT without keyword.

In addition to the above two components as core components, the CHINERIS system contains a core component, i.e., NE relation identification, which involves three units:

- *NE Relation Annotation*

  Based on the results of NE identification of the CHINERS system, this unit annotates NEs and their relations in texts for the consequent unit, that is, it inputs the results of NE identification and transforms the format of the results into XML format. At the same time, it adds NE relation (NER) tags in the annotated texts by interactive with users.

- *NE Relation Learning*

  It completes several tasks of positive and negative case-based learning, including (i) parsing XML trees and transforming their format into internal data format; (ii) building NER and non-NER pattern libraries; (iii) based on these two libraries, calculating average feature similarity, NER and non-NER self-similarity; (iv) after that, the feature sets (GCF's and ICF's sets) for NER and non-NER are determined; (v) then the corresponding feature weights and relation identification thresholds for NER and non-NER are determined as well.

- *NE Relation Identification*

  This unit is to identify different NERs using learning results. It inputs the sentence groups to be identified and constructs the corresponding candidate relation patterns. Then it uses NER and non-NER patterns from the libraries, Sports Ontology, as well as feature weights and identification thresholds provided by machine learning to match these candidate relations. Finally, it decides optimal identification tradeoff, resolves relation conflicts, infers omitted relations, and outputs identified results.

Notice that in Figure 8.1 the relevant learning results mean the selected features (GCFs and ICFs), the corresponding feature weights, and identification thresholds for different NERs or non-NERs.

In the next three subsections, we will elaborate the definition for each class as well as invoked relationships and executed sequence of methods in the classes for three core components.

[33] The other three named entities, namely, personal name (PN), date or time (DT), and location name (LN), are immediately identified after error repairing.

8.2.2 Word Segmentation and POS Tagging Processing

As we presented previously, an error repairer has been implemented in the CHINERS system. Its main functions include: (i) generating candidate rules to repair errors; (ii) selecting effective candidate rules; (iii) deciding final regular rules; (iv) sorting them; (v) rectifying related errors; (vi) counting repaired error numbers. Among these functions, with the help of machine learning, functions (i), (ii), and (iii) are fulfilled. To implement these functions, we defined nine classes in the *package*[34] MLForErrorReparation:

MLForErrorReparation **(main class)**: during training, it invokes the methods in classes CountErrorNum, GenerateRules to complete the generation of repair rules and statistics of the error number from word segmentation and POS tagging in original texts.

CountErrorNum: this class counts the error number of word segmentation and POS tagging.

GenerateRules: it generates the rules of rectifying word segmentation and POS tagging errors.

MLForSegRulesSelection **(stand-alone class[35])**: after the candidate rules of rectifying word segmentation errors are trained, it decides final effective rules in terms of their error repaired number in training texts.

MLForPOSRulesSelection **(stand-alone class)**: same as above for the candidate rules of rectifying POS tagging errors.

MLForRulesSorting **(stand-alone class)**: this class is used to sort the repair rules of word segmentation and POS tagging. Thus, it ensures that the rules that repair more errors in training texts can be early used early when testing.

RectifyErrorsWithContext: according to the repair rules with context constraints for word segmentation (including *concat*, *split*, and *slide* rules) and POS tagging, it repairs their corresponding errors.

RectifyErrorsWithoutContext: it repairs word segmentation and POS tagging errors, depending on the repair rules without context constraints for word segmentation and POS tagging.

[34] A package is a collection of related classes and interfaces providing access protection and namespace management in the Java programming language. To make classes easier to find and to use, to avoid naming conflicts, and to control access, programmers bundle groups of related classes and interfaces into packages.

[35] A stand-alone class means the methods of this class are only called by the methods in the class itself.

MLForRectifyingError: during the testing, it calls the methods in classes CountErrorNum, RectifyErrorsWithContext and RectifyErrorsWithoutContext to rectify errors using the rules with or without context constraints and count the remaining error number of word segmentation and POS tagging after repair.

In order to explain the invoked relationships between methods of classes as well as their executed sequence, we briefly specify them in Figure 8.2. Note that these relationships and sequence are only represented by classes rather than their methods. Thus, it is able to avoid falling into nitty-gritty details. In addition to that, it is stipulated that the executed sequence in the figure is from top to bottom and from left to right.

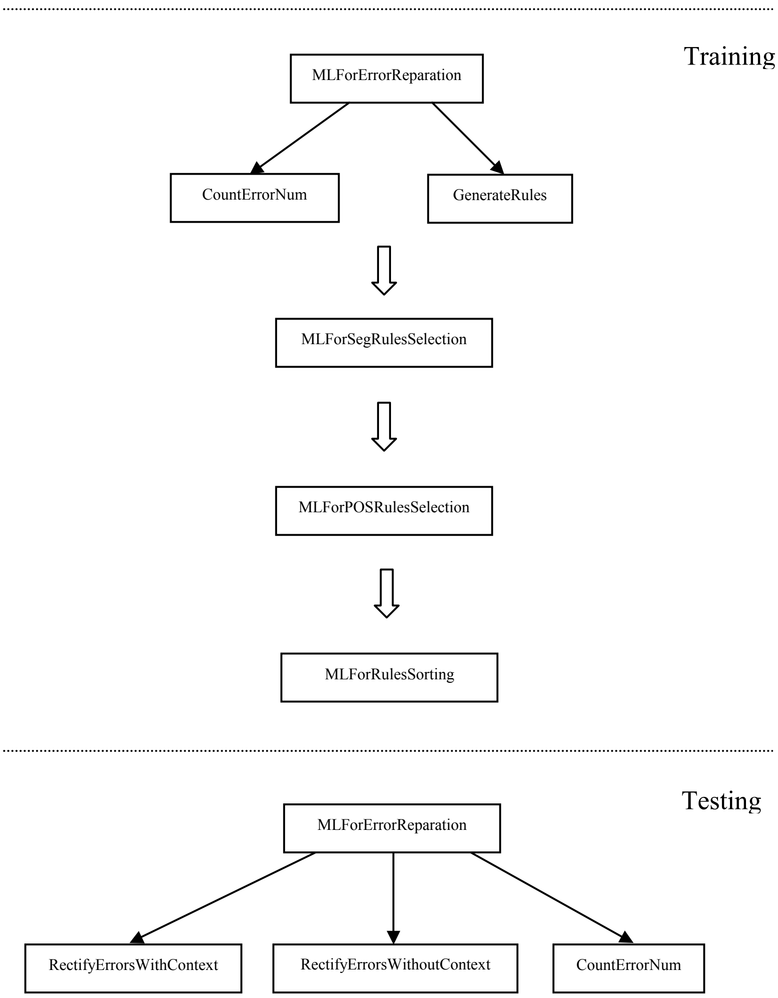


Figure 8.2 Invoked Relationships and Executed Sequence of Methods in the Classes for the Package MLForErrorReparation

Here it should also be observed that the error repair rules for word segmentation have been used *prior to* those for POS tagging; further, the error repair rules with context constraints enjoy priority in usage to those without context constraints.

8.2.3 Named Entity Identification

In the package NamedEntityIdentification, we implement three principal functions for NE identification: (i) automatically construct FSC according to the recognition rule sets for TN, CT, and PI; (ii) using the FSC, identify TN, CT, and PI from bottom to top level in turn, in which TN and CT have keywords such as 队 (Team) and 赛 (Competition); and (iii) for TN and CT with special constructions, identify them by a special approach (see Section 6.4). For the above functions, we defined 20 classes whose purposes are illustrated as follows:

***NamedEntityIdentification* (main class)**: during NE identification, it invokes the methods in classes OpenAndReadFile, SegmentRuleSet, SegPOSTagTransform, SegmentSentence, IdentifyTNWK, and IdentifyCTWK to input resources and sentences, construct FSC, identify corresponding NEs by the FSC, and identify TN and CT without keywords (special constructions).

OpenAndReadFile: this class is to input related resources and texts for NE identification.

SegmentRuleSet: it gets a rule from a rule set and calls the methods of the class ConstructFSC to construct FSC.

ConstructFSC: this is a main class for automatically constructing FSC. With each use, it builds one level of FSC, i.e., a recognizer, whose order is from low to high level.

GraphADT: it is a base class that provides a number of methods to establish directional graphs using adjacent matrixes.

AddRule: it adds a new recognition rule into the FSC being constructed. Before doing that, it must examine the legality of inserting an edge into a directional graph.

RuleInAdjMatrix: it checks if the same recognition rule already exists in FSC.

SegPOSTagTransform: this class transforms the format of the input file accomplishing word segmentation and POS tagging to the internal file format with text, paragraph, sentence, word, and punctuation tags.

SegmentSentence: it cuts a sentence from a text and invokes the methods of the classes MatchTNFSC, MatchCTFSC, and MatchPIFSC to identify TN, CT, and PI respectively.

MatchTNFSC: it loads related semantic information (gazetteers, etc.) into input sentences for matching TN in FSC.

MatchCTFSC: it puts corresponding semantic information (gazetteers etc.) into input sentences for matching CT in FSC.

MatchPIFSC: it supplies associated semantic information (gazetteers etc.) into input sentences for matching PI in FSC.

IdentifyNamedEntities: this is the class to identify TN, CT, and PI using the FSC mechanism. It is called by MatchTNFSC, MatchCTFSC, and MatchPIFSC classes.

IdentifyTNWK: sometimes there are some TNs with special constructions in sentences. This class is responsible for identifying such TNs.

CheckTNBalance: to keep the equity of domain verbs, it examines whether TN candidates in both sides of a domain verb maintain the equilibrium.

CheckTNContext: for the sake of utilizing the context clues of TN candidates, this class seeks whether there is a TN that is equal to the current TN candidate with the keyword in the whole text.

CheckTNAbbreviation: it checks if the current TN candidate is an abbreviation of TN.

IdentifyCTWK: it is analogous to the function of the class IdentifyTNWK, but for the identification of CTs with special constructions.

CheckCTContext: same as CheckTNContext class, but is for the identification of CT with special constructions.

CheckCTAbbreviation: similar to the CheckTNAbbreviation class, but for CT abbreviations.

The described provisions of the invoked relationships and executed sequence of the methods in the classes in Figure 8.3 are the same as the last figure. Note that because of limited figure space, the invoked relationships and executed sequence for IdentifyTNWK and IdentifyCTWK classes are expressed below the main part in the figure. In addition, the class MatchPIFSC is executed after the classes IdentifyTNWK and IdentifyCTWK are executed.

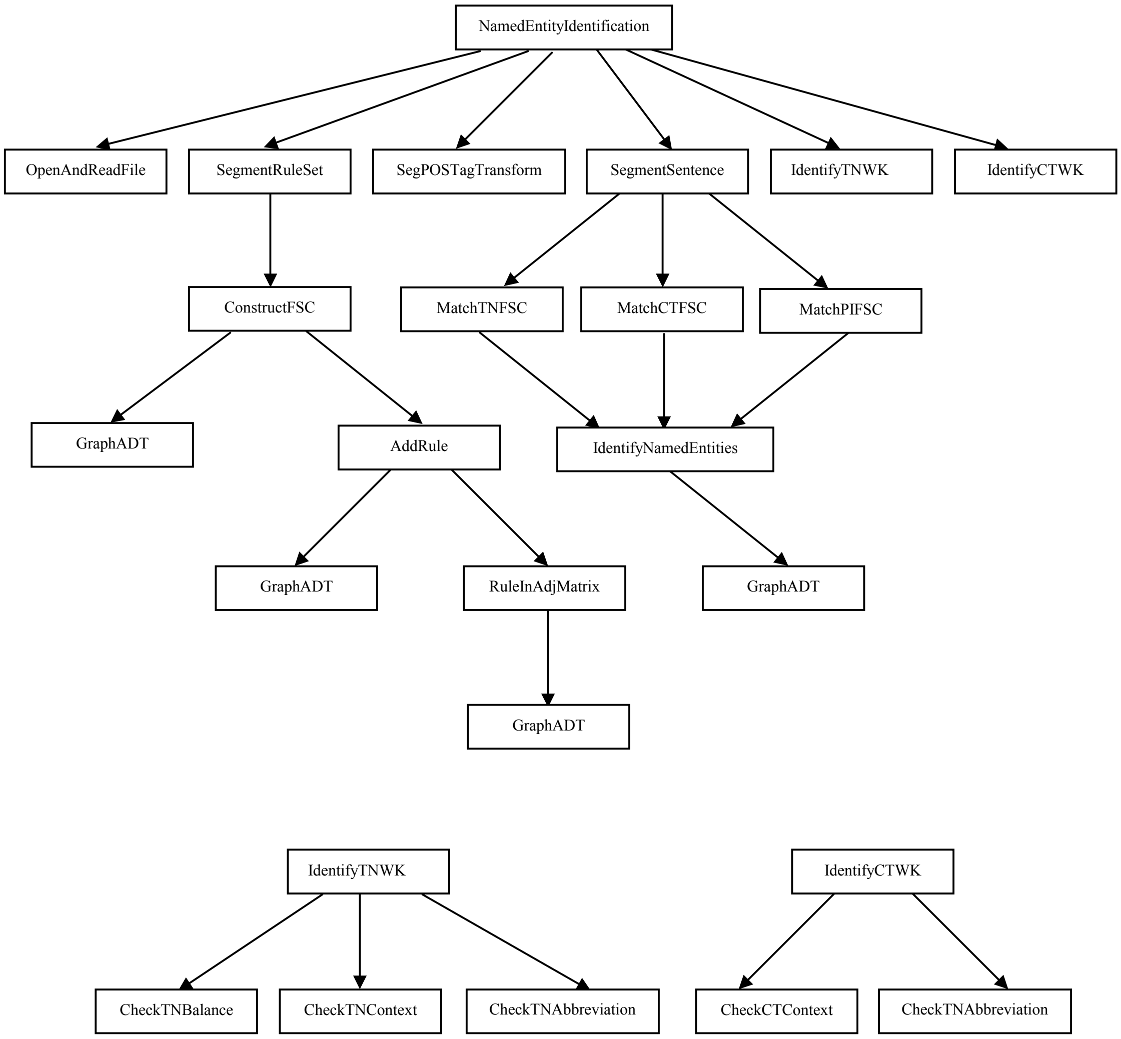


Figure 8.3 Invoked Relationships and Executed Sequence of Methods in the Classes for the Package NamedEntityIdentification

8.2.4 Named Entity Relation Identification

Based on positive and negative case-based learning and identification, the MLForRelationIdentification package including 24 classes is explained as follows. They accomplish two main functions: The first one is to learn NER and non-NER patterns, involving selecting important features, computing feature weights, and determining identification thresholds for different NERs and non-NERs; the second one is to identify 14 different NERs, embodying calculating the similarity between candidate relations and NER / non-NER patterns, deciding optimal identification tradeoff, resolving NER conflicts, and inferring omitted NERs.

AnnotationXML: it inputs the result of NE identification and transforms the format of the result into XML format. At the same time, it adds NER tags in the annotated text by interactive with user.

Relation: it transforms the type codes of NER into NER type representations.

DomTreeNERelationTextTransform: it is used to transfer XML files into the internal data format having word, NE and NER tags, etc.

BuildRelationAndNonRelationPatterns: it is used to build NER and non-NER patterns for NER identification.

SetFeatureAndWeightForRelation: based on the NER pattern library, it calculates average feature similarity and NER self-similarity. After that it picks up the GCFs for NERs and calculates the involved feature weights and NER identification thresholds.

InputRelationOrNonRelationPattern: this class is used to input NER or non-NER patterns for positive and negative case-based machine learning.

SentenceTypeSimilarity: it computes the similarity of sentence types. Two compared relations lie in the corresponding sentence.

NESenPosTypeSimilarity: this class is responsible for calculating the similarity of the sentence position type. The NEs related to two matched relations exist in the sentences.

NEOrderSimilarity: it counts the similarity of the NE order for two related relations.

NEVerbPositionSimilarity: the class carries out the computation of the similarity of NE-verb relative position for two associated relations.

NEContextSimilarity: this class conducts the calculation of the similarity for the context of NEs dealing with two involved relations.

VerbSentenceSimilarity: it completes the calculation of the similarity for two matched NERs depending on verb position(s) in sentence(s).

NEAndParticleOrderSimilarity: it is used to compute the similarity of the order between NE(s) and particle(s) for two compared relations.

NEPOSSimilarity: this class is to count the similarity of POS of NEs involved in two related relations.

NEContextPOSSimilarity: the class carries out the computation of the similarity in terms of the POS context of NEs associated with two associated relations.

SentencePOSSimilarity: it executes the calculation of the similarity of POS of two sentences having relevant relations.

ValenceSimilarity: according to the verb valence whose arguments (i.e., NE) relate to two compared relations, this class computes their similarity.

NEConceptSimilarity: it calculates the similarity of NE concepts for two matched relations.

VerbConceptSimilarity: based on verb concepts, this class carries out the computation of the similarity.

DetermineClassRelationship: the class is to determine the corresponding class hierarchical relationship of two instances in Sports Ontology.

SetFeatureAndWeightForNonRelation: according to non-NER pattern library, it calculates average feature similarity and non-NER self-similarity. Then it selects the ICFs for non-NERs and computes the corresponding feature weights and non-NER identification thresholds.

BuildIdentifiedNERelationPatterns: for the sake of the unification of computing data, this class is to build the relation patterns for identified relations.

GetClassInstance: it accesses Sports Ontology and obtains class instances.

NERelationIdentification: This class inputs the sentence groups to be identified and their candidate relation patterns. Furthermore, it uses NER and non-NER patterns, Sports Ontology, as well as feature weights and identification thresholds to match these candidate relations. Finally, it decides optimal identification tradeoff, resolves relation conflicts, infers omitted relations, and outputs identified results.

In Figure 8.4, there are 13 above-mentioned classes (from the class SentenceTypeSimilarity to the class VerbConceptSimilarity in turn) participating in computing different similarity for matched relations during training and testing procedures, in which a method of the class VerbConceptSimilarity calls the methods of the class DetermineClassRelationship for fixing the involved class hierarchical relationship of two instances in Sports Ontology. In addition, because of similar operations, the classes SetFeatureAndWeightForRelation and SetFeatureAndWeightForNonRelation in the training procedure are merged into a square frame of the class.

The system implementation can be primarily divided into two phases: in the first phase, we implemented the CHINERS system which improves the quality of word segmentation and POS tagging and conducts the identification of NEs (including NEs with special constructions); then, in the second phase, we accomplished another core component for the identification of NERs and integrated it with the CHINERS system into a new system, i.e., the CHINERIS system. The system shows that the run-time efficiency is acceptable. The system user interfaces are friendly (see Appendix F) as well.

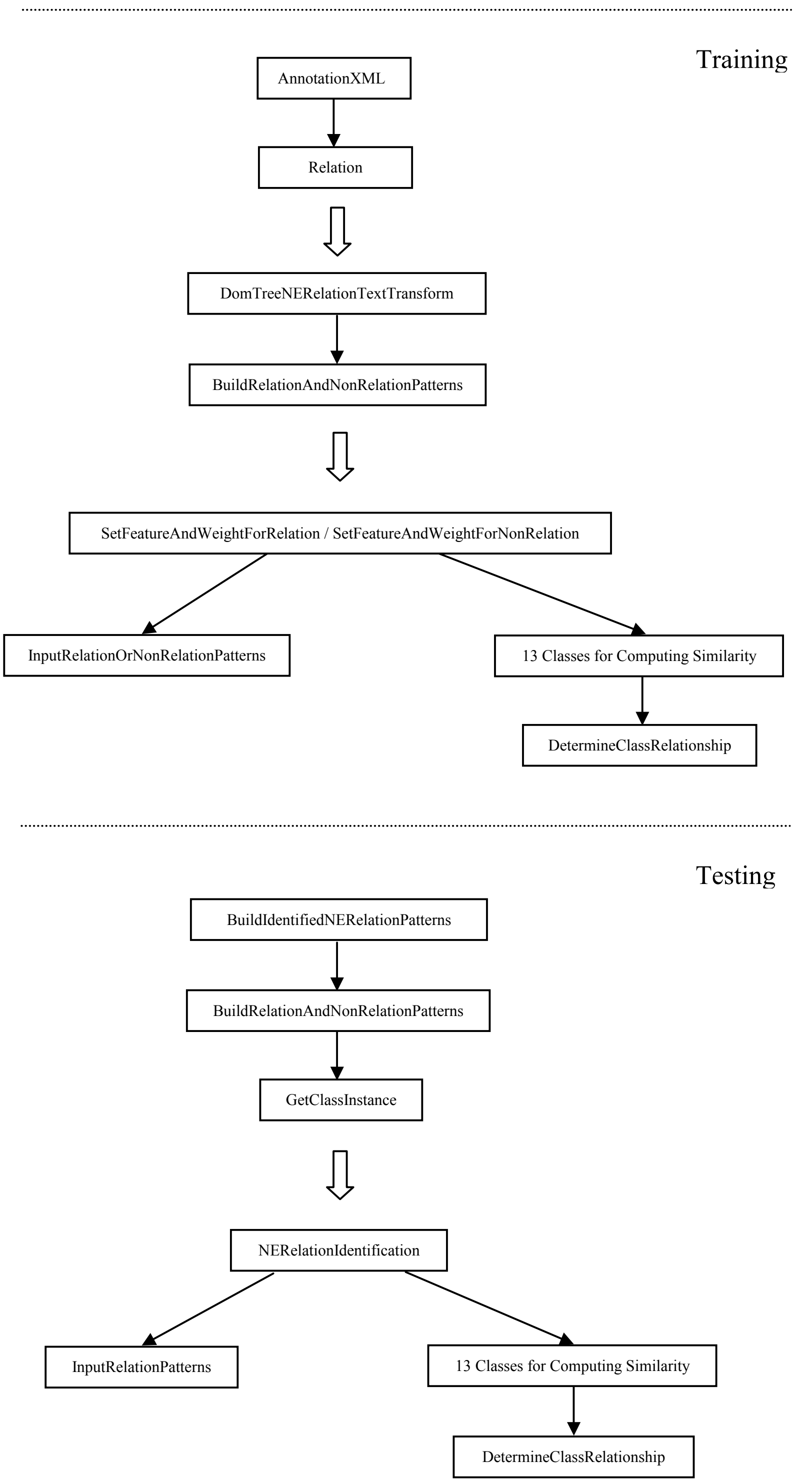

Training
AnnotationXML
Relation
DomTreeNERelationTextTransform
BuildRelationAndNonRelationPatterns
SetFeatureAndWeightForRelation / SetFeatureAndWeightForNonRelation
InputRelationOrNonRelationPatterns
13 Classes for Computing Similarity
DetermineClassRelationship
Testing
BuildIdentifiedNERelationPatterns
BuildRelationAndNonRelationPatterns
GetClassInstance
NERelationIdentification
InputRelationPatterns
13 Classes for Computing Similarity
DetermineClassRelationship

Figure 8.4 Invoked Relationships and Executed Sequence of Methods in the Classes for the Package MLForRelationIdentification

8.3 Discussion

In this chapter, we have presented the architecture of the CHINERIS system and the implementation methods of its core components. During the implementation, some beneficial strategies have been adopted such as:

- Object-oriented design and development;
- Integrate available application systems and resources, e.g., Modern Chinese Word Segmentation and POS Tagging System, Protégé-2000, and HowNet;
- Adopt the Java programming language, so that the system can be run on different platforms;
- Give priority to the development of basic classes in Java, early-developed classes provide the support for late-developed classes when debugging.

Because of these strategies, the development time of the system is saved and the system reliability is enhanced. Additionally, the system architecture is reasonable and the system can be independent of the running environment.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

In this thesis, we have lucubrated IE approaches applied to Chinese texts in a specific domain. In order to availably identify Chinese named entities (NEs) and named entity relations (NERs), we propose a Chinese IE computational model with three stages, i.e., word processing, NE identification, and NER identification.

For guaranteeing the *quality* of Chinese word processing, a machine learning method - transformation-based error-driven machine learning has been employed in the error repair for word segmentation and POS tagging in the first stage. Because the text domain in our research is *unique*, it is suitable and effective to apply such an approach to error repair. In addition, compared to developing a new word segmentation and POS tagging system, it requires *less* development work. In error repair, adopting context-sensitive and context-insensitive error repair rules sequentially, we can attend to both *individual-character* and *general-character* errors and rectify more errors. It is a beneficial exploration to *simultaneously* enhance the performance of Chinese word segmentation and POS tagging, and this exploration has achieved *positive* results as well.

Generally speaking, the identification of NEs and NERs might need an ontology to provide domain knowledge. Considering the actual requirements for domain knowledge, we defined a hierarchical taxonomy and constructed conceptual relationships among Object, Movement and Property concept categories under the taxonomy in Sports Ontology. Thus, this ontology can play a *crucial* role in the identification of NEs and NERs, such as the recognition of NEs with *special constructions* - without trigger words, the determination of *NE boundaries*, and the provision of *feature values* as well as the computation of the *semantic distance* for two concepts during the identification of NERs.

In the second stage, we utilize *Finite State Cascades* (*FSC*) as a shallow parser to identify different NEs. An approach that can automatically construct FSC relying upon recognition rules is suggested. In this approach, we theoretically extend the original definition of Finite State Automata (FSA), in other words, we use *multiple constraint* symbols rather than atomic constraint symbols. With this extension, we improve the *practicability* for the FSC mechanism. At the same time, the new issue for automatically constructing FSC also increases the *flexibility* of its maintenance. For processing special linguistic phenomena which *cannot* be recognized by FSC, with the help of Sports Ontology, a strategy for the identification of NEs without trigger words is added to this stage to improve the identification performance.

The work in the third stage is principal and difficult in our investigation, because the defined NERs are fairly different and they are allowed across sentences. A proposal concerning an innovative learning and identifying approach for NERs called *positive and negative case-based learning and identification* is offered. This approach pursues the improvement of the identification performance for NERs through learning two *opposite* types of cases (NER and non-NER patterns) simultaneously, automatically selecting effectual *multi-level* linguistic features from a predefined feature set for each NER and non-NER, and making an *optimal* identification tradeoff.

The goal of the learning is to capture *valuable* information from NER and non-NER patterns. This information is implicated in different features and helps us identify NERs and non-NERs. In general, because not all features we predefine are important for each NER or non-NER, we should select them by a *reasonable* measure mode. Depending on 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 corresponding pattern library, the effective feature sets - *general-character feature* (*GCF*) sets for NERs and *individual-character feature* (*ICF*) sets for non-NERs are established. Moreover, the GCF and ICF *feature weighting* serve as a proportional determination of feature importance degree for identifying NERs or non-NERs. At the same time, the *identification thresholds* for them 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 at the same time, both positive and negative cases. In such situations, we have to employ *voting* to decide which existing case environment is more similar to this new case. In addition, a number of *special circumstances* such as relation conflicts and relation omission should be considered during the identification. Therefore, two corresponding strategies resolving relation conflicts and inferring missing relations, are applied to the identification procedure for improving the identification performance.

This thesis work includes not only the investigation of computational model, but also the implementation of the prototype system. The workload of coding for the prototype system is extremely heavy (approximately 15, 000 lines of Java code), however, we can utilize this system for experiments that can confirm the *realizability* and *validity* for proposed approaches.

In the research, some lessons also exist that are worth reconsidering. As mentioned previously, it is still imperfect for Chinese morphological, syntactic and semantic theories. Hence, when tackling the actual problems, we have to work out some strategies relying upon the *actual needs*, which perhaps have no versatility. Besides, this is to underestimate the *engineering workload* for the prototype system, especially for Sports Ontology and the experimental data statistics. In this way, it is difficult to have the schedule of research work in hand. Another lack is the difficultly of finding an *analogous* Chinese IE system (particularly an NER identification system) that can directly carry on the comparison with our experimental results. Hence, the *shortage* of a comprehensive evaluation for this model.

In spite of the insufficiencies in this thesis, I hope that the efforts of this thesis can be beneficial to status of the current research for Chinese IE.

9.2 Future Work

In the experiments, the *rationality* and *validity* of the approaches adopted in the Chinese IE computational model have been proved. However, there are still some points in need of improvement in the future work:

- Because of the great number of repair rules generated for improving the performance of word segmentation and POS tagging, there possibly exist *rule conflicts*, e.g., there are two or more different repair results under the same context condition, and implicit *rule redundancy*, such as some rules can be inferred by other rules within the rule libraries. Therefore, a component to automatically optimize repair rules, comprising the resolution of rule conflicts and removal of implicit rule redundancy should be considered;

- Currently, the recognition rule sets are *manually* crafted for each class of NEs based on the observations of domain linguistic phenomena. However, this procedure consumes a lot of labour and time. A machine learning approach to automatic acquisition of NE recognition rules (Soderland et al., 1995; Califf et al., 1999) can settle this problem;
- Similarly, the gathering task of information from various sources for Sports Ontology also takes a lot of labour and time in our research. Therefore, the investigation of *automatically* acquiring information for ontologies is very important (Uschold and Gruninger, 1996; Zhou and Feng, 2000; Ngai et al., 2002; Wu and Hsu, 2002);
- In the research of NER identification, we find that personal and demonstrative pronouns also are *critical* features, which can help us to identify NERs. For that reason, we can consider adding a new IE task in the model, namely, *coreference identification* (Brennan et al., 1987; Chen, 1993; Grosz et al., 1995; Ng and Cardie, 2002).

Analogously to IE's theories and technologies applied to other languages, Chinese IE certainly also has *bright* prospects, regardless of technologies and real-world applications.

Appendix A

Some Examples of Chinese Morphology, Grammar, and Semantics

A.1 Morphology

(i) Simple Words

Example A.1.1 (unbroken word)

枇杷 (loquat; pí pá[36]) and 骆驼 (camel; luò tuo); The first word is an alliterated unbroken word whose two initials are all p. The second one is a rhymed unbroken word whose two finals are the same, that is, uo. Of course, there exist unbroken words that are neither alliterated nor rhymed. e.g., 芙蓉 (hibiscus; fú róng)

Example A.1.2 (phone-reduplicate word)

奶奶 (grandmother; nǎi-nai), 区区 (trivial; qū-qu), and 尝尝 (taste … a little; cháng-chang). Most of such words are syntactically and/or semantically distinct from the single morphemes. For example, the original morpheme 奶 means breasts, milk (noun) or suckle (verb); but the reduplicate word 奶奶 means grandmother. Therefore, if the phone-reduplicate word is only represented by one syllable, it is meaningless.

Example A.1.3 (transliterated word)

咖啡 (coffee; kā fēi), 迪斯科 (disco; dí sī ke), and 奥林匹克 (Olympics; ào lín pǐ kè). Most polysyllabic simple words are transliterated words.

Example A.1.4 (onomatopoetic word)

扑通 (splash; pū tōng), 轰隆隆 (rumble; hōng lōng lōng), and 唧唧喳喳 (chirp or chatter; jī jī zhā zhā).

(ii) Compound Words

[36] This is the phonetic spelling for the Chinese word 枇杷.

Example A.1.5 (some commonly used prefixes)

老 (lǎo), 小 (xiǎo), 第 (dì), 初 (chū), 可 (kě) and so on. 老 and 小 are typically prefixed to people's surnames to form nicknames. 第 is added to numerals to form ordinal numbers. 初 is added to the numerals one to ten to denote the first ten days of a lunar month in China. 可 can be put in front of a number of verbs to form an adjective. Its meaning may be described as '-able'.

Example A.1.6 (some infixes)

得 (obtain; de), 不 (not; bu) etc. The only forms that could be considered infixes in Chinese occur with resultative verb compounds, which are always composed of two elements, the second one signals some result of the action or process conveyed by the first one. Note that when 得 and 不 are inserted into such compounds, the meaning of the compound is opposite, e.g., 说得清楚 (say + obtain + clear = can say clearly) and 说不清楚 (say + not + clear = cannot say clearly).

Example A.1.7 (some common suffixes)

儿 (er), 们 (men), 学 (xué), 家 (jiā), 化 (huà), 子 (zi), 头 (tou) etc. 儿 is a retroflex suffix and the only non-syllabic suffix in Chinese. It merges with the preceding syllable to form a new syllable ending in the retroflex sound, such as, 鸟儿 (bird, niǎor). 们 is pronounced with a neutral tone and is restricted to nouns and pronouns which refer to humans only. With this suffix, the single form of a noun changes to its plural form. However, a monosyllabic noun does not take this plural suffix. For instance, the word 老师们 (teachers) is correct, but the word 兵们 (soldiers) is unacceptable. 们 can also be a plural marker for pronouns. 学 is the Chinese counterpart of the English suffix '-ology'; while 家 is equivalent to the English suffix '-ist', for example, 动物学 (zoology) and 动物学家 (zoologist). 化 creates verbs from nouns and adjectives. It is semantically equivalent to the English suffix '-ize', e.g. 工业化 (industrialize). Etymologically, 子 is a suffix derived from the morpheme 'child'. It always has the neutral tone, and it constitutes the obligatory second syllable of a large number of nouns when they occur as independent words, such as, 梯子 (ladder). Another neutral-tone syllable that is a suffix is 头, which occurs in modern Chinese with a number of nouns that are *bound morphemes* that are allowed to attach a morpheme. For instance, 骨头 (bone).

(iii) Construction Modes for Compound Words

a) Complex Forms

Example A.1.8 (coordination)

道路 (road + way = road), 开关 (turn on + turn off = switch), 手足 (hand + foot = brothers).

Example A.1.9 (verb-object)

刺眼 (offend + eye = dazzling).

Example A.1.10 (modification)

黑板 (black + board = blackboard), 葡萄酒 (grape + wine = grape wine).

Example A.1.11 (subject-predicate)

地震 (earth + quake = earthquake), 人造 (man + make = man-made).

Example A.1.12 (predicate-complement)

说明 (say + clear = explain), 推翻 (push + turn over = overthrow).

Example A.1.13 (reduplication)

爸爸 (father), 常常 (often) etc.

b) Adjunctive Forms

Example A.1.14 (prefix + root)

老师 (teacher): The meaning of this prefix 老 is indistinct;

老大 (the eldest child): The prefix 老 precedes the morphemes that represent the seniority among brothers and sisters;

老张 (Lao Zhang): Here this prefix 老 is added in front of people’s surnames. The compound word denotes a respectful form of address for acquaintances.

阿姨 (aunt): In general, the prefix 阿 is added in front of relative names or some single-syllable given names.

Example A.1.15 (root + suffix)

桌子 (desk or table): Basically, the suffix 子 follows a noun. The compound word denotes the name and description of a thing;

猫儿 (cat): This suffix 儿 is added to some nouns, so that the formed compound word carries some emotional color;

盼头 (hope): The suffix 头 is added to some verbs. The compound word gives the name and description of a thing;

茫然 (baffled, bewildered): This suffix 然 is a mark of adjective or adverb.

A.2 Grammar

(i) Combination Phrases of Full Words

Example A.2.1 (coordination phrase)
兄弟姐妹 (brothers + sisters = brothers and sisters)

Example A.2.2 (verb-object phrase)
美化环境 (beautify + environment = beautify the environment)

Example A.2.3 (modification phrase)
木头房子 (wood + house = wooden house)

Example A.2.4 (subject-predicate phrase)
病情发展 (patient’s condition + development = development of patient’s condition)

Example A.2.5 (predicate-complement phrase)
听清楚 (listen + clear = listen clearly)

Example A.2.6 (apposition phrase)
中国首都北京 (Chinese + capital + Beijing = Chinese capital Beijing)

(ii) Combination Phrases between Full Words and Particles

Example A.2.7 (quantifier phrase)
二十个 (twenty + GE = twenty); Here, GE is a quantifier.

Example A.2.8 (locative phrase)
小河边 (creek + close by = by a creek)

Example A.2.9 (proposition phrase)
通过事实调查 (through + facts + scrutinizing = through scrutinizing the facts)

Example A.2.10 (auxiliary phrase)
他们的 (they + DE = their) ;

牛奶似的 (milk + SHI DE = milky);

所接受 (SUO + acceptance = what is accepted)

Note: DE, SHI DE and SUO are all *structure auxiliary words*. DE is used after an attribute and indicates that the linguistic unit following it is an object being modified by the attribute. SHI DE is used after nouns, pronouns and verbs and indicates that the preceding word is similar to something or some condition. When SUO is used before a verb, they are combined into a nominal structure.

Example A.2.11 (volitive phrase)

可能发生 (possible + occur = possible to occur)

Example A.2.12 (directional phrase)

奔过去 (rush + over = rush)

(iii) Sentence Patterns

Example A.2.13 (N Sentence)

咖啡，糖，牛奶。 (coffee, sugar, and milk)

关于她妈妈的故事。 (The story about her mother.)

Example A.2.14 (AN Sentence)

屋子里净人。(There are nothing but people in the room.) Here, 净 (nothing but) is an adverb that modifies the noun 人 (people).

Example A.2.15 (AP Sentence)

挡不住的诱惑。 (enticement without obstruction) Note that 挡不住 (without obstruction) is an attribute which modifies the predicate 诱惑 (enticement) with the help of the structure auxiliary word 的 (DE).

Example A.2.16 (P Sentence)

再见！(good-bye!)

心情沉重地思索了很久。(Deeply think very long with a heavy heart.)

Example A.2.17 (SP Sentence)

雨停了，天晴了。 (The rain stops and it is clear.)

两支球队将要交锋。(Two teams will compete.)

我们把脏衣服都洗了。(We have washed all the dirty clothes.) Note that in the last sentence given above, the proposition 把 (BA) leads an adverbial phrase 把脏衣服 (BA + dirty clothes) which is actually the object of the predicate 洗 (wash).

Example A.2.18 (VO Sentence)

开饭了。(Meal is ready.) In this sentence, 开 (ready) is a verb and 饭 (meal) is an object;

随手关门。(Close the door behind one.) Here, 随 (along with) and 关 (close) are two verbs; while 手 (hand) and 门 (door) are two objects.

Example A.2.19 (SVO Sentence)

我们参加了这次比赛。(We have participated in this competition.)

大家感到高兴。(Everybody feels happy.)

Example A.2.20 (SVOO Sentence)

警察罚了那个违章者五元钱。(The policeman has fined that regulation-breaker five Yuans.) In this sentence, two objects can be combined with the verb 罚 (fine) independently. Such as, 警察罚了那个违章者。(The policeman has fined that regulation-breaker.) or 警察罚了五元钱。(The policeman has fined five Yuans.)

你浪费了我整整一天的时间。(You have wasted me one day’s time.) Here, the verb 浪费 (waste) cannot only collocate with the near-object 我 (me).

王先生告诉我一个好消息。(Mr. Wang tells me good news.) In this sentence, the verb 告诉 (tell) cannot collocate with the far-object 消息 (news) alone.

Example A.2.21 (VOO Sentence)

再给我一个机会吧！(Please give me another chance.)

A.3 Semantics

(i) Semantic Architecture

Example A.3.1 (Modern Chinese Kinship Terminologies)

In modern Chinese kinship terminologies, close kinship terminologies include 妻子 (wife), 丈夫 (husband), 父 (father), 母 (mother), 子(son), 女 (daughter), 兄 (elder brother), 弟 (younger brother), 姐 (elder sister) and 妹 (younger sister); the second-level kinship terminologies mean 祖父 ((paternal) grandfather), 外祖父 ((maternal) grandfather), 祖母 ((paternal) grandmother), 外祖母 ((maternal) grandmother), 孙子 (son’s son), 外孙 (daughter’s son), 孙女 (son’s daughter), 外孙

女 (daughter's daughter), 公公 (husband's father), 岳父 (wife's father), 婆婆 (husband's mother), 岳母 (wife's mother), 大伯子 (husband's elder brother), 小叔子 (husband's younger brother), 大舅子 (wife's elder brother), 小舅子 (wife's younger brother), 姐夫 (elder sister's husband), 妹夫 (younger sister's husband), 大姑子 (husband's elder sister), 小姑子 (husband's younger sister), 大姨子 (wife's elder sister), 小姨子 (wife's younger sister), 嫂子 (elder brother's wife), 弟媳妇 (younger brother's wife), 女婿 (daughter's husband) and 儿媳妇 (son's wife); the part of the third level kinship terminologies are 伯父 (father's elder brother), 叔父 (father's younger brother), 舅父 (mother's brother), 姑父 (the husband of father's sister), 姨父 (the husband of mother's sister), 姑母 (father's sister(married)), 姨母 (mother's sister), 伯母 (the wife of father's elder brother), 婶母 (the wife of father's younger brother), 舅母 (wife of mother's brother), 侄子 (brother's son), 外甥 (sister's son), 侄女 (brother's daughter) and 外甥女 (sister's daughter).

In the *vertical* direction in Figure A.1, the architecture has not been completed (see the ellipsis symbols). The sub-semantic fields need to be analyzed further. But close kinship terminologies (parent sub-semantic field) have clear relationships with spouse relation, bearing relation and compatriot relation of kinship terminologies (children sub-semantic fields). These three children sub-semantic fields have their fixed sememes. The sememes of spouse relation terminologies are wife and husband, the ones of bearing relation terminologies include father, mother, son and daughter, and the ones of compatriot relation terminologies are composed of elder brother, younger brother, elder sister and younger sister. Their parent sub-semantic field, the close kinship terminologies, however, has no sememe which only belongs to itself. The sememes under this category are the sum of the sememes under its children sub-semantic fields.

In the *horizontal* direction in Figure A.1, there are two different cases: there might not exist clear and sensible points in which the three children sub-semantic fields of close kinship terminologies are different from each other; while the common kinship terminologies are obviously classified according to the distance of the relationship.

(ii) Semantic Structure of Sentence

a) Topic and Comment

Example A.3.2

苏先生买了一辆自行车。(Mr. Su bought a bicycle). In this sentence, "Mr. Su" is a topic and "bought a bicycle" is a comment.

b) Constituents of Sentence Semantic Structure

Example A.3.3 (predicate with zero argument)

明天入伏。(The hottest part of the summer will start tomorrow.) Here, 明天 (tomorrow) is an adverbial; 入伏 (the hottest part of the summer starts) is a predicate.

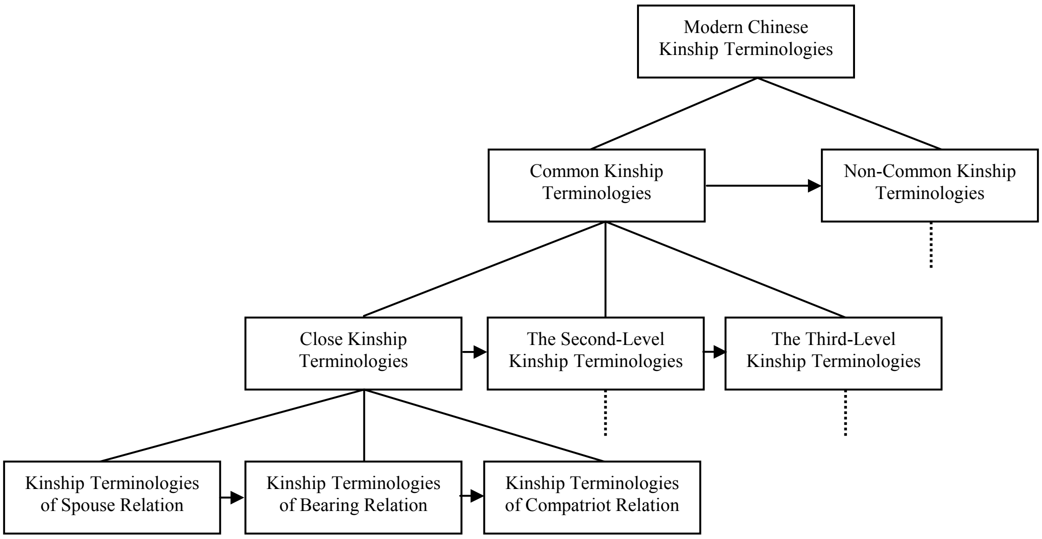

Modern Chinese Kinship Terminologies
Common Kinship Terminologies
Non-Common Kinship Terminologies
Close Kinship Terminologies
The Second-Level Kinship Terminologies
The Third-Level Kinship Terminologies
Kinship Terminologies of Spouse Relation
Kinship Terminologies of Bearing Relation
Kinship Terminologies of Compatriot Relation

Figure A.1 Semantic Structure of Modern Chinese Kinship Terminologies

Example A.3.4 (predicate with one argument)

雨后的颐和园真美。(After rain, the summer palace is really beautiful.) In the sentence, 颐和园 (the summer palace) is an argument.

Example A.3.5 (predicate with two arguments)

闵先生有辆摩托车。(Mr. Min has a motor.) In the sentence, 闵先生 (Mr. Min) and 摩托车 (motor) are two arguments.

Example A.3.6 (predicate with three arguments)

这位售货员递给王小姐一双鞋。(This shop assistant hands over a pair of shoes to Miss Wang.) Here, 售货员 (shop assistant), 王小姐 (Miss Wang) and 鞋 (shoes) are three arguments.

Example A.3.7 (nominative: agent)

孩子们[37]都站在门外。(The children are all standing outside the door.)

Example A.3.8 (nominative: possession)

李白是唐代的诗人。(Bai Li is a poet of the Tang Dynasty.)

Example A.3.9 (nominative: participation)

[37] The underlining of the word 孩子们 means it is an agent.

除了三班的，其他的代表都到了。(Except for the representative of class three, all the others have arrived.)

Example A.3.10 (nominative: experience)

昨天太兴奋，夜里我失眠了。(Yesterday I was too aroused, and I couldn't sleep at night.)

Example A.3.11 (accusative: patient)

杯子让明明打碎了。(The cup was broken by Ming Ming.)

Example A.3.12 (accusative: result)

小鸟搭窝。 (A small bird built its nest.)

Example A.3.13 (accusative: declaration)

刘老师是初二三班的班主任。(Teacher Liu is the class adviser of Class three, Grade two in the junior middle school.)

Example A.3.14 (accusative: object)

我在颐和园碰见了几个老战友。(I met several former comrade-in-arms in the summer palace.)

Example A.3.15 (dative)

保险公司付了向家一大笔钱。(The insurance company has paid a large amount of money to the Xiang Family.)

Example A.3.16 (circumstance: range)

他俩的兴趣在数学上，一个喜欢代数，一个喜欢几何。(They both are interested in mathematics. One prefers algebra, and the other likes geometry more.)

Example A.3.17 (circumstance: time)

过去我们就住在这里。(In the past we lived here.)

Example A.3.18 (circumstance: space)

e.g. 这件事传遍了山下的大小村庄。(This thing spreads over the small and big villages under the mountain.)

Example A.3.19 (means: tool)

老李用大碗吃饭。(Lao Li has a meal with a big bowl.)

Example A.3.20 (means: material)

这个小区的住户一开始就用煤气做饭。(The residents in this neighbourhood cooked with gas from the very beginning.)

Example A.3.21 (means: manner)

笔试合格后，还要面试。(After passing the written examination, you still need to pass an oral quiz.)

Example A.3.22 (means: scale)

寿比南山 (Wish your life lasts as long as the South Mountain!)

Example A.3.23 (reason: basis)

根据合同我们要扣你的奖金。(In accordance with the contract, we want to deduct your bonus.)

Example A.3.24 (reason: cause)

这场足球赛让小楚伤透了心。(Because of this football match Xiao Chu was really heart-broken.)

Example A.3.25 (reason: goal)

为了健康要天天锻炼。(In order to keep fit, we should have some physical exercise every day.)

Example A.3.26 (modification: genitive)

北京的烤鸭 (roast Beijing duck)

Example A.3.27 (modification: description)

儿科大夫 (pediatric doctor)

Example A.3.28 (modification: apposition)

e.g. 古都洛阳 (ancient capital Luoyang)

Example A.3.29 (modifier for predicate: mode or modality)

懒洋洋地坐着 (lounge on a chair)

Example A.3.30 (modifier for predicate: time or space)

院子里坐 (sit in the courtyard)

Example A.3.31 (modifier for predicate: degree)

格外高兴 (especially cheerful)

Example A.3.32 (modifier for predicate: range)

只有大米 (There is only rice.)

Example A.3.33 (modifier for predicate: negation)

别讨论 (Don’t discuss.)

Example A.3.34 (modifier for predicate: repeat)

再三嘱咐 (advise again and again)

Example A.3.35 (modifier for predicate: mood)

居然责备 (unexpectedly blame)

Example A.3.36 (complement constituent of predicate: result)

衣服晾干了 (The clothes dried.)

Example A.3.37 (complement constituent of predicate: time or space)

来自五湖四海 (from all corners of the country)

Example A.3.38 (complement constituent of predicate: tool or material)

乘火车 (take a train)

Example A.3.39 (complement constituent of predicate: possibility)

说得清楚 (can explain something clearly)

Example A.3.40 (complement constituent of predicate: degree)

站得整整齐齐 (stand in perfect order)

Example A.3.41 (complement constituent of predicate: quantity of action or behavior)

算了两遍 (have counted twice)

Example A.3.42 (linking constituent)

外面下着大雪，屋里却很暖和。(It is snowing heavily outside, but it is quite warm in the room.). Here, 却 (but) is a linking constituent and expresses a contrastive relationship.

Appendix B

Examples of Error Repair Rules

B.1 Error Repair Rules for Word Segmentation

(i) POS Context Constraints

a) Concat

rectify_segmentation_error(concat, 中|F|远|A, 1, N, 上海|N5, 汇|N)

rectify_segmentation_error(concat, 伊|R|普斯|N5|维奇|N7, 2, N7, 小胜|N5, ，|W)

b) Split

rectify_segmentation_error(split, 韩日|N7, 1, J|J, 进军|N, 的|U)

rectify_segmentation_error(split, 本周日和|N5, 1|3, R|DT|C, 参加|V, 阿曼|N7)

c) Slide

rectify_segmentation_error(slide, 大|A|连队|N, right1, N5|N, _|_, 主帅|N)

rectify_segmentation_error(slide, 其中|F|场|W, left1, R|N, 耳|N, 巴斯图|N5)

(ii) Without Context Constraints

a) Concat

rectify_segmentation_error(concat, 伤兵|N|满|A|营|N, 2, I)

rectify_segmentation_error(concat, 国|N|足|N, 1, J)

b) Split

rectify_segmentation_error(split, 江苏队|N4, 2, N5|N)

rectify_segmentation_error(split, 中场巴拉克|N5, 2, N|N7)

c) Slide

rectify_segmentation_error(slide, 申|N2|花国庆|N4, right1, N|N)

rectify_segmentation_error(slide, 佛罗|N5|伦萨胜威尼斯|N7, right2|left3, N7|V|N7)

B.2 Error Repair Rules for POS Tagging

(i) **POS Context Constraints**

rectify_tag_error (上三路|N5, N, ’|W, ’|W)

rectify_tag_error (沪|N, J, 到|P, 后|D)

(ii) **Without Context Constraints**

rectify_tag_error (新年|N, DT)

rectify_tag_error (打圆场|N, V)

Appendix C

Hierarchical Taxonomy of Sports Ontology

C.1 Hierarchical Taxonomy

The following is a suggested hierarchical taxonomy of Sports Ontology, in which we have implemented all of Movement concept categories and a part of Object and Property concept categories:

- **Object**
  - **Material Object**
    - **Material**
      - **Living Things**
        - **Human Beings**
          - **Society**
            - **Country**
              - **Province**
                - **Province City**
                  - **Province City Location**
              - **State**
                - **State City**
                  - **State City Location**
                - **Special City**
                  - **Special City Location**
              - **Abstract Country**
            - **Organization**
              - **Association**
              - **Committee**
              - **Sports Team**
              - **Rooting Section**
            - **Individual**
              - **Occupation**

- **Relationship**
- **Hobby**
- **Ability and Insight**
- **Other Types**
- **Collective**
  - **Ability and Insight**
- **Activity**
  - **Ball_Activity**
  - **Competition**
- **Non-Living Things**
  - **Natural Things**
  - **Artificial**
    - **Equipment**
    - **Knowledge Produces**
    - **Building**
- **Spirit Object**
  - **Sensation**
    - **Touch**
    - **Taste**
    - **Smell**
    - **Sight**
    - **Hearing**
  - **Feelings**
    - **Mood**
    - **Sympathy**
    - **Morals**
  - **Idea**
    - **View**
    - **Ideal**
    - **Stratagem**
  - **Act**
    - **Words**
    - **Behavior**
    - **Language**
  - **Disposition**
    - **Temper**

      - **Particular Character**
    - **Measure**
  - **Abstract Object**
    - **Business**
    - **Environment**
    - **Reason**
    - **Character**
    - **Performance**
    - **Illness**
    - **Measure**
  - **Time**
    - **Regular Time**
      - **Year**
        - **Seasons**
          - **Month**
            - **Week**
              - **Day**
                - **Day Division**
      - **Others**
    - **Intervalic Time**
  - **Space**
    - **Direction**
      - **Geography Direction**
      - **Reference Direction**
      - **Others**
    - **Position**
      - **Point**
      - **Line**
      - **Angle**
      - **Surface**
      - **Solid**
      - **Others**

- **Movement**
  - **Artificiality**
    - **Action**

    - **Grant**
    - **Direction**
    - **Motion**
      - **Attack**
      - **Defence**
      - **Transfer**
  - **Behavior**
    - **Thought**
      - **Desire**
      - **Thinking**
    - **Perception**
    - **Cognition**
      - **Perceive**
      - **Cognizance**
    - **Conduct**
      - **Admission**
      - **Order**
      - **Doing**
  - **Status**
    - **Dynamic Status**
      - **Existence**
      - **Possession**
      - **Classification**
    - **State**
      - **Relationship**
        - **Win and Lose**
        - **Draw**
        - **Others**
      - **Variation**
  - **Process**
    - **Express**
    - **Usage**
    - **Produce**
    - **Development**
- **Non-Artificiality**

- **Property**
  - **Appearance**
    - **Length**
    - **Height**
    - **Magnitude**
    - **Width**
    - **Depth**
    - **Thickness**
    - **Straight**
    - **Sharpness**
    - **Roughness**
    - **Cliffy**
    - **Shape**
    - **Figure**
  - **Presentation**
    - **Amount**
    - **Proportion**
    - **Density**
    - **Concise**
    - **Weight**
    - **Maturity**
    - **Sound**
    - **Brightness**
    - **Distance**
    - **Time**
    - **Speed**
    - **Period**
    - **Weather**
    - **Style**
    - **Looks**
    - **Quality**
    - **Age**
    - **Health**
    - **Frequency**
  - **Color**
    - **Simple**

  - **Combination**
- **Flavor**
  - **Smell**
  - **Taste**
- **Nature**
  - **Reality**
  - **Quality**
  - **Qualification**
  - **Grade**
  - **Difficulty**
  - **Comprehension**
  - **Integrality**
  - **Correction**
  - **Rationality**
  - **Flexibility**
  - **Acuity**
  - **Profundity**
  - **Chariness**
  - **Force**
  - **Glory**
  - **Notability**
  - **Nobility**
  - **Importance**
  - **Urgency**
  - **Necessity**
  - **Familiarity**
  - **Harmony**
  - **Compactness**
  - **Simplicity**
  - **Particularity**
  - **Universality**
  - **Loveliness**
  - **Devilishness**
  - **Laughably**
  - **Normality**
  - **Publicity**

- **Validity**
- **Distinctness**
- **Serviceability**
- **Advantage**
- **Aptitude**
- **Savageness**
- **Foreign**
- **Canonicity**
- **Public**
- **Kindred**
- **Master**
- **Reference**

Ability and Political Integrity

- **Goodness**
- **Honesty**
- **Faithfulness**
- **Blandness**
- **Optimism**
- **Harshness**
- **Pizazz**
- **Conciliation**
- **Determination**
- **Boldness**
- **Acumen**
- **Flyness**
- **Pliable**
- **Capability**
- **Eruditely**
- **Promising**
- **Foresight**
- **Tact**
- **Craftiness**
- **Active**
- **Diligence**
- **Persistence**
- **Economy**

- **Earnest**
- **Carefulness**
- **Sober**
- **Bovine**
- **Refinement**
- **Pure and Lofty**
- **Deferential**
- **Condescension**
- **Magnanimous**
- **Strict**
- **Democracy**
- **Dredge**
- **Equity**
- **Probity**
- **Nobleness**
- **Pudicity**
- **Amorous**

Circumstances

- **Peace and Tranquility**
- **Boom**
- **Jollification**
- **Quiet**
- **Lone**
- **Busyness**
- **Comfortable**
- **Safety**
- **Blessedness**
- **Meek**
- **Awfulness**
- **Cleanliness**
- **Orderliness**
- **Affluence**

Appendix D

Examples of Named Entity Recognition Rules

D.1 Recognition Rules for Team Name

$rule_1$: TN → J + KEY_WORD | AbbreviationCityName + TeamNameKeyword | AbbreviationClubName + TeamNameKeyword | AbbreviationCountryName + TeamNameKeyword | AbbreviationName + TeamNameKeyword | AbbreviationProvinceName + TeamNameKeyword

$rule_2$: TN → N + N + KEY_WORD | ClubName + TNOtherName + TeamNameKeyword | ClubName + ProductName + TeamNameKeyword | Range + TNOtherName + TeamNameKeyword

$rule_3$: TN → N7 + N + KEY_WORD | CityName + TNOtherName + TeamNameKeyword | CompanyName + CompanyName + TeamNameKeyword | CountryName + TNOtherName + TeamNameKeyword | ProvinceName + CompanyName + TeamNameKeyword | StateName + CityName + TeamNameKeyword

$rule_4$: TN → N1 + N + N + KEY_WORD | CountryName + TNOtherName + TNOtherName + TeamNameKeyword | ProvinceName + CityName + ProductName + TeamNameKeyword

$rule_5$: TN → N5 + N + QT + KEY_WORD | CityName + ProductName + AlphabeticalString + TeamNameKeyword

$rule_6$: TN → N1 + QT + M + N + KEY_WORD | CountryName + AlphabeticalString + NumericalString + TNOtherName + TeamNameKeyword

$rule_7$: TN → N + QT + M + N7 + N + KEY_WORD | TNOtherName + AlphabeticalString + NumericalString + CountryName + TNOtherName + TeamNameKeyword

$rule_8$: TN → N7 + M + M + M + N + N + N + KEY_WORD | CountryName + NumericalString + NumericalString + NumericalString + Quantifier + TNOtherName + TNOtherName + TeamNameKeyword

D.2 Recognition Rules for Competition Title

rule_1: CT → N + KEY_WORD | AbbreviationName + CompetitionTitleKeyword | CTOther Name + CompetitionTitleKeyword | Range + CompetitionTitleKeyword

rule_2: CT → J + J + KEY_WORD | AbbreviationName + AbbreviationName + Competition TitleKeyword | AbbreviationCountryName + AbbreviationCountryName + CompetitionTitleKeyword

rule_3: CT → J + N + M + KEY_WORD | CTOtherName + CTOtherName + NumericalString + CompetitionTitleKeyword

rule_4: CT → M + DT + N + J + KEY_WORD | NumericalString + Quantifier + CTOther Name + AbbreviationName + CompetitionTitleKeyword

rule_5: CT → A + N + N + M + QT + KEY_WORD | Range + CTOtherName + CTOther Name + Rank + AlphabeticalString + CompetitionTitleKeyword

rule_6: CT → J + J + J + M + N + N + KEY_WORD | AbbreviationCountryName + AbbreviationCountryName + AbbreviationCountryName + NumericalString + Quantifier + CTOtherName + CompetitionTitleKeyword

rule_7: CT → H + M + Q + N7 + N + N + J + KEY_WORD | Prefix + NumericalString + Quantifier + CityName + CTOtherName + CTOtherName + AbbreviationName + CompetitionTitleKeyword

rule_8: CT → M + W + M + DT + N7 + N + M + N + KEY_WORD | NumericalString + PunctuationMark + NumericalString + CTOtherName + CountryName + CTOtherName + Rank + Quantifier + CompetitionTitleKeyword

rule_9: CT → H + M + Q + N + M + N + A + N + N + KEY_WORD | Prefix + Numerical String + Quantifier + ContinentName + NumericalString + Quantifier + Range + CTOtherName + CTOtherName + CompetitionTitleKeyword

D.3 Recognition Rules for Personal Identity

rule_1: PI → N + N4 | PersonalTitle + PersonalName

rule_2: PI → N + N7 | PersonalTitle + PersonalName

rule_3: PI → N5 + N | CityName + PersonalTitle

rule_4: PI → M + Q + N | NumericalString + Quantifier + PersonalTitle

rule_5: PI → TN + KEY_WORD + N | TeamName + TeamNameKeyword + PersonalTitle

rule_6: PI → TN + KEY_WORD + N + N4 | TeamName + TeamNameKeyword + Personal Title + PersonalName

rule_7: PI → TN + KEY_WORD + U + N | TeamName + TeamNameKeyword + Auxiliary Word + PersonalTitle

Appendix E

An Example of the Self-Similarity Calculation, Feature Selection, Feature Weight Computing and Identification Threshold Determination

E.1 Two Sentence Groups and Their Relation Patterns

Sentence Group$_1$ (SG$_1$)[38]:

本 报 讯 中国 女足 昨天 在 悉尼 进行 的 泛 太平洋 国际 女足 邀请赛 第 二 轮 角逐 中 ， 被 东道主 澳大利亚 队 1 比 1 逼平 。

Newspaper news: In the second round match of the International Invitational Tournament for Womens Football for the Pacific Ocean Zone, which was held in Sydney yesterday, the Chinese Womens Football Team was compelled to draw by 1:1 by the host Australia Team.

Sentence Group$_2$ (SG$_2$):

本 报 讯 即将 在 一周 后 出战 欧锦赛 的 南斯拉夫 国家 足球队 ， 昨天 在 这里 怎么 也 想不到 会 以 2 ： 4 败 于 中国 香港 的 南华 俱乐部 队 。

Newspaper news: after one week, the Yugoslavia National Football Team, which will soon go to battle in the European Championships, did not expect that yesterday it would have been defeated here by the Nanhua Club Team from Hongkong, China, by 2:4.

Relation Pattern of SG$_1$:

[38] In order to *clearly* enumerate Chinese words in sentence groups, we add a *space* between words to represent word *boundaries* in sentences. In addition, the *underlines* of words indicate that *named entities* are composed of these words.

no = 1

RE = {(TM_CP, NE1-1, NE1-4), (HT_VT, NE2-2, NE1-1), (DT_DT, NE1-1, NE2-2), (CP_TI, NE1-4, NE1-2), (CP_LOC, NE1-4, NE1-3), (TM_CP, NE2-2, NE1-4), (ID_TM, NE2-1, NE2-2)}

SC = {(1, 本, native/one's_own/central, Empty, aValue/attachment/self), (2, 报, bulletin/report/cable, Empty, document/letter/publications), (3, 讯, dispatch/information/message, Empty, information), (6, 在, at/in/on, Empty, Vgoingon/location/scope), (8, 进行, be_on_the_march/march/be_in_progress, Empty, GoForward/GoOn/Vgoingon), (9, 的, empty, AuxiliaryWord, DeChinese), (11, 第, -th, Prefix, aValue/sequence/ordinal), (12, 二, 2/two, NumericalString, qValue/amount/cardinal/mass), (13, 轮, round, Empty, NounUnit/event), (14, 角逐, contend/enter_into_rivalry/tussle, Empty, compete/fight), (15, 中, center/middle/China, Empty, location/middle/time/now), (16, ，, ,, Empty, {punc}), (17, 被, empty, Empty, LeChinese), (20, １, 1, NumericalString, qValue/amount/cardinal/mass), (21, 比, compare, Empty, CompareTo), (22, １, 1, NumericalString, qValue/amount/cardinal/mass), (23, 逼平, force_to_draw, Empty, force/ResultEvent=equal/sport), (24, 。, ., Empty, {punc})}

st = multi-sentences

NE = {(NE1-1, 4, TN, {(1, 中国), (2, 女足)}), (NE1-2, 5, Time, {(1, 昨天)}), (NE1-3, 7, LN, {(1, 悉尼)}), (NE1-4, 10, CT, {(1, 泛), (2, 太平洋), (3, 国际), (4, 女足), (5, 邀请赛)}), (NE2-1, 18, PI, {(1, 东道主)}), (NE2-2, 19, TN, {(1, 澳大利亚), (2, 队)})}

NEC = {(NE1-1, 讯, 昨天), (NE1-2, 女足, 在), (NE1-3, 在, 进行), (NE1-4, 的, 第), (NE2-1, 被, 澳大利亚), (NE2-2, 东道主, １) }

VERB = {(8, 进行), (14, 角逐), (23, 逼平)}

PAR = {(6, 在), (9, 的), (13, 轮), (17, 被), (21, 比)}

NEP = {(NE1-1, {(1, N1), (2, J)}), (NE1-2, {(1, N)}), (NE1-3, {(1, N7)}), (NE1-4, {(1, A), (2, N), (3, N), (4, J), (5, N)}), (NE2-1, {(1, N)}), (NE2-2, {(1, N7), (2, N)})}

NECP = {(NE1-1, N, N), (NE1-2, J, P), (NE1-3, P, V), (NE1-4, U, H), (NE2-1, U, N7), (NE2-2, N, M)}

SP = {(1, A), (2, N), (3, N), (4, NE1-1), (5, NE1-2), (6, P), (7, NE1-3), (8, V), (9, U), (10, NE1-4), (11, H), (12, M), (13, Q), (14, V), (15, N), (16, W), (17, U), (18, NE 2-1), (19, NE2-2), (20, M), (21, P), (22, M), (23, V), (24, W)}

VV = {(V_8, {Agent|fact/compete|CT, -Time|time|DT}), (V_14, {Agent|human/mass|TN}), (V_23, {Agent|human/mass|TN, Patient|human/mass|TN})}

(Continued)

NECT = {(NE1-1, place/country/ProperName/Asia+fact/exercise/sport/female), (NE1-2, time/past/day), (NE1-3, place/city/ProperName/Australia), (NE1-4, aValue/content/Empty/undesired+waters/surfacial/ProperName+community/country+fact/exercise/sport/female+fact/compete/sport), (NE2-1, human/entertain), (NE2-2, place/country/ProperName/Australia+community/human/mass)}

VCT = {(V_8, GoForward/GoOn/Vgoingon), (V_14, compete/fight), (V_23, force/ResultEvent=equal/sport)}

Relation Pattern of SG$_2$:

no = 2

RE = {(CP_TI, NE1-3, NE1-1), (CP_TI, NE1-3, NE1-2), (TM_CP, NE1-4, NE1-3), (TM_CP, NE2-4, NE1-3), (WT_LT, NE2-4, NE1-4), (TM_CPC, NE2-4, NE2-2), (LOC_CPC, NE2-3, NE2-2), (TM_CPC, NE2-4, NE2-3)}

SC = {(1, 本, native/one's_own/central, Empty, aValue/attachment/self), (2, 报, bulletin/report/cable, Empty, document/letter/publications), (3, 讯, dispatch/information/message, Empty, information), (5, 在, at/in/on, Empty, {Vgoingon}/{location}/{scope}), (7, 后, after_a_certain_time/Hou/queen, Empty, time/future/surname), (8, 出战, go_to_war, Empty, engage/content=fight), (10, 的, empty, AuxiliaryWord, {DeChinese}), (12, ，, Empty, Empty, Empty), (14, 在, at/in/on, Empty, {Vgoingon}/{location}/{scope}), (15, 这里, here/this_place, Empty, location/special), (16, 怎么, how, Empty, {manner/question}), (17, 也, also/as_well/either, Empty, also), (18, 想不到, unexpect, Empty, ThinkOf/^$predict), (19, 会, be_able_to/can/be_skillful_in, Empty, BeAble/ComeTogether/meet), (20, 以, according_to/because_of/by_means_of, Empty, {AccordingTo}/{cause}/{means}), (21, ２, 2, NumericalString, qValue/amount/cardinal/mass), (22, ：, :, Empty, {punc}), (23, ４, 4, NumericalString, qValue/amount/cardinal/mass), (24, 败, be_defeated/spoil/wither, Empty, defeated/damage/decline), (25, 于, than/at/in, Empty, {contrast}/{location}/{patient}), (28, 的, empty, AuxiliaryWord, {DeChinese}), (30, 。, ., Empty, {punc})}

st = multi-sentences

NE = {(NE1-1, 4, Time, {(1, 即将)}), (NE1-2, 6, Time, {(1, 一), (2, 周)}), (NE1-3, 9, CT, {(1, 欧锦赛)}), (NE1-4, 11, TN, {(1, 南斯拉夫), (2, 国家), (3, 足球), (4, 队)}), (NE2-1, 13, Time, {(1, 昨天)}), (NE2-2, 26, LN, {(1, 中国)}), (NE2-3, 27, LN, {(1, 香港)}), (NE2-4, 29, TN, {(1, 南华), (2, 俱乐部), (3, 队)})}

(Continued)

NEC = {(NE1-1，讯，在), (NE1-2，在，后), (NE1-3，出战，的), (NE1-4，的，，), (NE2-1，，，在), (NE2-2，于，香港), (NE2-3，中国，的), (NE2-4，的，。)}

VERB = {(8，出战), (18，想不到), (19，会), (24，败)}

PAR = {(5，在), (10，的), (14，在), (16，怎么), (17，也), (20，以), (25，于), (28，的)}

NEP = {(NE1-1, {(1, N)}), (NE1-2, {(1, M), (2, N)}), (NE1-3, {(1, J)}), (NE1-4, {(1, N7), (2, N), (3, N), (4, N)}), (NE2-1, {(1, N)}), (NE2-2, {(1, N1)}), (NE2-3, {(1, N5)}), (NE2-4, {(1, N5), (2, N), (3, N)})}

NECP = {(NE1-1, N, P), (NE1-2, P, N), (NE1-3, V, U), (NE1-4, U, W), (NE2-1, W, P), (NE2-2, P, N5), (NE2-3, N1, U), (NE2-4, U, W)}

SP = {(1, A), (2, N), (3, N), (4, NE1-1), (5, P), (6, NE1-2), (7, N), (8, V), (9, NE1-3), (10, U), (11, NE1-4), (12, W), (13, NE2-1), (14, P), (15, N), (16, D), (17, D), (18, V), (19, V), (20, P), (21, M), (22, W), (23, M), (24, V), (25, P), (26, NE2-2), (27, NE2-3), (28, U), (29, NE2-4), (30, W)}

VV = {(V_8, {Agent|human/mass|TN, Patient|fact/compete|CT, -Time|time|DT, -Location|facilities/@exercise/sport|LN}), (V_18, {Agent|human/mass|TN, Agent|human/*look/#entertainment/#sport/*recreation|PN}), (V_19, {Agent|human/mass|TN}), (V_24, {Agent|human/mass|TN, Patient|human/mass|TN})}

NECT = {(NE1-1, duration/TimeShort), (NE1-2, qValue/amount/cardinal+ character/surname/human/ProperName), (NE1-3, place/ProperName+fact/compete/sport), (NE1-4, place/country/ProperName/(Europe)+place/#human/country/politics+SportTool/fact/exercise+community/human/mass), (NE2-1, time/past/day), (NE2-2, place/country/ProperName/(Asia)), (NE2-3, place/city/ProperName/(China)), (NE2-4, Empty+ InstitutePlace/@associate/@recreation/#literature/#entertainment/#sport+community/human/mass)}

VCT = {(V_8, engage/content=fight), (V_18, Empty), (V_19, BeAble/ComeTogether/meet), (V_24, defeated/damage/decline)}

E.2 Calculation of the Self-Similarity for a Relation

In the last section, the relation patterns corresponding to SG_1 and SG_2 have been listed. In order to calculate the *self-similarity* of the relation CP_TI, first of all, we define the relations CP_TI_1, CP_TI_{21}, and CP_TI_{22} by the NER general frame (see Definition 7.2) as follows:

CP_TI$_1$(泛太平洋国际女足邀请赛 (International Invitational Tournament for Female Football in the General Pacific Ocean Zone), 1-1-4; 昨天 (yesterday), 1-1-2);

CP_TI$_{21}$(欧锦赛 (European Championships), 1-1-3; 即将 (soon), 1-1-1);

CP_TI$_{22}$(欧锦赛 (European Championships), 1-1-3; 一周 (one week), 1-1-2).

Utilizing the calculation formulas (7.5) - (7.18) in Section 7.3.3, the similarity of computing results for every feature among the above three relations is given in the following:

Sim (CP_TI$_1$, CP_TI$_{21}$) (STF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NESPF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NEOF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NEVPF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NECF) = 0 + 0.5 / 2 = 0.25

Sim (CP_TI$_1$, CP_TI$_{21}$) (VSPF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NEEWPOF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NEPF) = (0 / 1 + 1 / 1) / 2 = 0.5

Sim (CP_TI$_1$, CP_TI$_{21}$) (NECPF) = 0 + 0.5 / 2 = 0.25

Sim (CP_TI$_1$, CP_TI$_{21}$) (SPF) = (4+3+2) / 10 = 0.9

Sim (CP_TI$_1$, CP_TI$_{21}$) (VVF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (NECTF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{21}$) (VCTF) = max (0.8 * 1.6 / (5 + 1.6) + 0.2 * 0, 0.8 * 1.6 / (0 + 1.6) + 0.2 * 0.5) = 0.9

Sim (CP_TI$_1$, CP_TI$_{22}$) (STF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NESPF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NEOF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NEVPF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NECF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (VSPF) = 1.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NEEWPOF) = 0.25

Sim (CP_TI$_1$, CP_TI$_{22}$) (NEPF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NECPF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (SPF) = (4+3+2) / 10 = 0.9

Sim (CP_TI$_1$, CP_TI$_{22}$) (VVF) = 0.0

Sim (CP_TI$_1$, CP_TI$_{22}$) (NECTF) = 1.0

Sim (CP_TI_1, CP_TI_{22}) (VCTF) = max (0.8 * 1.6 / (5 + 1.6) + 0.2 * 0, 0.8 * 1.6 / (0 + 1.6) + 0.2 * 0.5) = 0.9

Sim (CP_TI_{21}, CP_TI_{22}) (STF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (NESPF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (NEOF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (NEVPF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (NECF) = 0 + 0.5 = 0.5

Sim (CP_TI_{21}, CP_TI_{22}) (VSPF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (NEEWPOF) = 0.5

Sim (CP_TI_{21}, CP_TI_{22}) (NEPF) = (0 / 1 + 1 / 1) / 2 = 0.5

Sim (CP_TI_{21}, CP_TI_{22}) (NECPF) = 0 + 0.5 = 0.5

Sim (CP_TI_{21}, CP_TI_{22}) (SPF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (VVF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (NECTF) = 1.0

Sim (CP_TI_{21}, CP_TI_{22}) (VCTF) = 1.0

Based on the above results and the formulas (7.2) and (7.3) defined in Section 7.3.3, the similarity of three relations and the self-similarity of the CP_TI is calculated in turn:

$$\text{Sim}(CP\_TI_1, CP\_TI_{21}) = \frac{\sum_{1}^{13} \text{Sim}(CP\_TI_1, CP\_TI_{21})(f_t)}{13} = \frac{7.8}{13} = 0.60$$

Similarly,

$$\text{Sim}(CP\_TI_1, CP\_TI_{22}) = \frac{7.05}{13} = 0.54$$

$$\text{Sim}(CP\_TI_{21}, CP\_TI_{22}) = \frac{11}{13} = 0.85$$

$$\mathrm{Sim}_{\mathrm{average}}(\mathrm{CP\_TI}) = \frac{\sum_{1}^{3} \mathrm{Sim}\,(\mathrm{CP\_TI}_{\,j},\ \mathrm{CP\_TI}_{\,k})}{\mathrm{Sum}_{\mathrm{relation\_pair}}(\mathrm{CP\_TI}_{\,j},\ \mathrm{CP\_TI}_{\,k})} = \frac{0.60 + 0.54 + 0.85}{3} = 0.66$$

Note: In the above calculation formulas, i = 12 (this relation no. denotes the relation CP_TI); m = 3; $1 \le j, k \le m, j \ne k$; $\mathrm{Sum}_f = 13$.

E.3 Feature selection, feature weight computing and identification threshold determination

As we mentioned in Section 7.3.4, there are two kinds of features in every relation, i.e., *general-character* and *individual-character features* (*GCFs* and *ICFs*). The suggested *feature selection standard* depends on the *self-similarity* of a relation. In this example, the self-similarity of the relation CP_TI is $\mathrm{Sim}_{\mathrm{average}}$(CP_TI). If the average similarity value of a feature is greater than the value of $\mathrm{Sim}_{\mathrm{average}}$(CP_TI), it is classified as a GCF; otherwise, it is an ICF. According to the self-similarity value (0.66) of the relation CP_TI, the GCFs of this relation are: STF, NESPF, NEOF, VSPF, SPF, NECTF, VCTF; while its ICFs are: NEVPF, NECF, NEEWPOF, NEPF, NECPF, VVF. Thus, we can use the above GCFs to identify CP_TI relations and ICFs to recognize non-CP_TI relations.

Depending on the calculation formulas (7.19) and (7.20), the *feature weights* for each GCF feature are counted:

$$f(s)_w(R(i)) = \frac{\mathrm{Sim}_{\mathrm{average}} f(s)(R(i))}{\sum_{t=1}^{n} \mathrm{Sim}_{\mathrm{average}} f(t)(R(i))}$$

$$\begin{aligned}\sum_{t=1}^{n} \mathrm{Sim}_{\mathrm{average}} f(t)(R(i)) = \sum_{t=1}^{7} \mathrm{Sim}_{\mathrm{average}} f(t)(R(i)) &= \mathrm{Sim}_{\mathrm{average}}\mathrm{STF}(\mathrm{CP\_TI}) + \mathrm{Sim}_{\mathrm{average}}\mathrm{NESPF} \\ &\quad (\mathrm{CP\_TI}) + \mathrm{Sim}_{\mathrm{average}}\mathrm{NEOF}(\mathrm{CP\_TI}) + \\ &\quad \mathrm{Sim}_{\mathrm{average}}\mathrm{VSPF}\,(\mathrm{CP\_TI}) + \mathrm{Sim}_{\mathrm{average}}\mathrm{SPF} \\ &\quad (\mathrm{CP\_TI}) + \mathrm{Sim}_{\mathrm{average}}\mathrm{NECTF}(\mathrm{CP\_TI}) + \\ &\quad \mathrm{Sim}_{\mathrm{average}}\mathrm{VCTF}(\mathrm{CP\_TI}) \\ &= 1.0 + 1.0 + 1.0 + 1.0 + 0.93 + 1.0 + 0.93 \\ &= 6.86\end{aligned}$$

According to the above calculation formula for $f(s)_w$:

$$STF_w(CP\_TI) = \frac{Sim_{average}STF(CP\_TI)}{\sum_{t=1}^{7} Sim_{average}f(t)(CP\_TI)} = \frac{1.0}{6.86} = 0.145772594$$

Analogously,

$$NESPF_w(CP\_TI) = \frac{1.0}{6.86} = 0.145772594$$

$$NEOF_w(CP\_TI) = \frac{1.0}{6.86} = 0.145772594$$

$$VSPF_w(CP\_TI) = \frac{1.0}{6.86} = 0.145772594$$

$$SPF_w(CP\_TI) = \frac{0.93}{6.86} = 0.135568513$$

$$NECTF_w(CP\_TI) = \frac{1.0}{6.86} = 0.145772594$$

$$VCTF_w(CP\_TI) = \frac{0.93}{6.86} = 0.135568513$$

Next, the *identification threshold* for the relation CP_TI is calculated by the calculation formula (7.21):

$$\text{IdenThrh(CP\_TI)} = \frac{\sum_{t=1}^{7} \text{Sim}_{\text{average}}f(t)(\text{CP\_TI})}{7} = \frac{6.86}{7} = 0.98$$

E.4 Explanation of the Calculation Results

1) Each feature similarity can *objectively* express the analogous degree between two NERs from an aspect, e.g., in the above example, Sim (CP_TI$_{21}$, CP_TI$_{22}$) (f$_t$) is greater than or equal to Sim (CP_TI$_1$, CP_TI$_{21}$) (f$_t$) and Sim (CP_TI$_1$, CP_TI$_{22}$) (f$_t$). That means the analogous degree between CP_TI$_{21}$ and CP_TI$_{22}$ is *closer* than any other two pairs of relations.

2) The similarity of relations is characterized by the average similarity of their features. Obviously, Sim (CP_TI$_{21}$, CP_TI$_{22}$) is greater than Sim (CP_TI$_1$, CP_TI$_{22}$) and Sim (CP_TI$_1$, CP_TI$_{22}$) as well.

3) The self-similarity of a relation indicates the *concentrative degree* of this kind of relation, i.e., the *average distance* among all pairs of the relation instances. If this value is lower, obviously, these kind of relations are relatively different from each other (the average distance is longer); conversely, they are similar (the average distance is shorter). It is *appropriate* that the self-similarity of a relation serves as the selection standard for features.

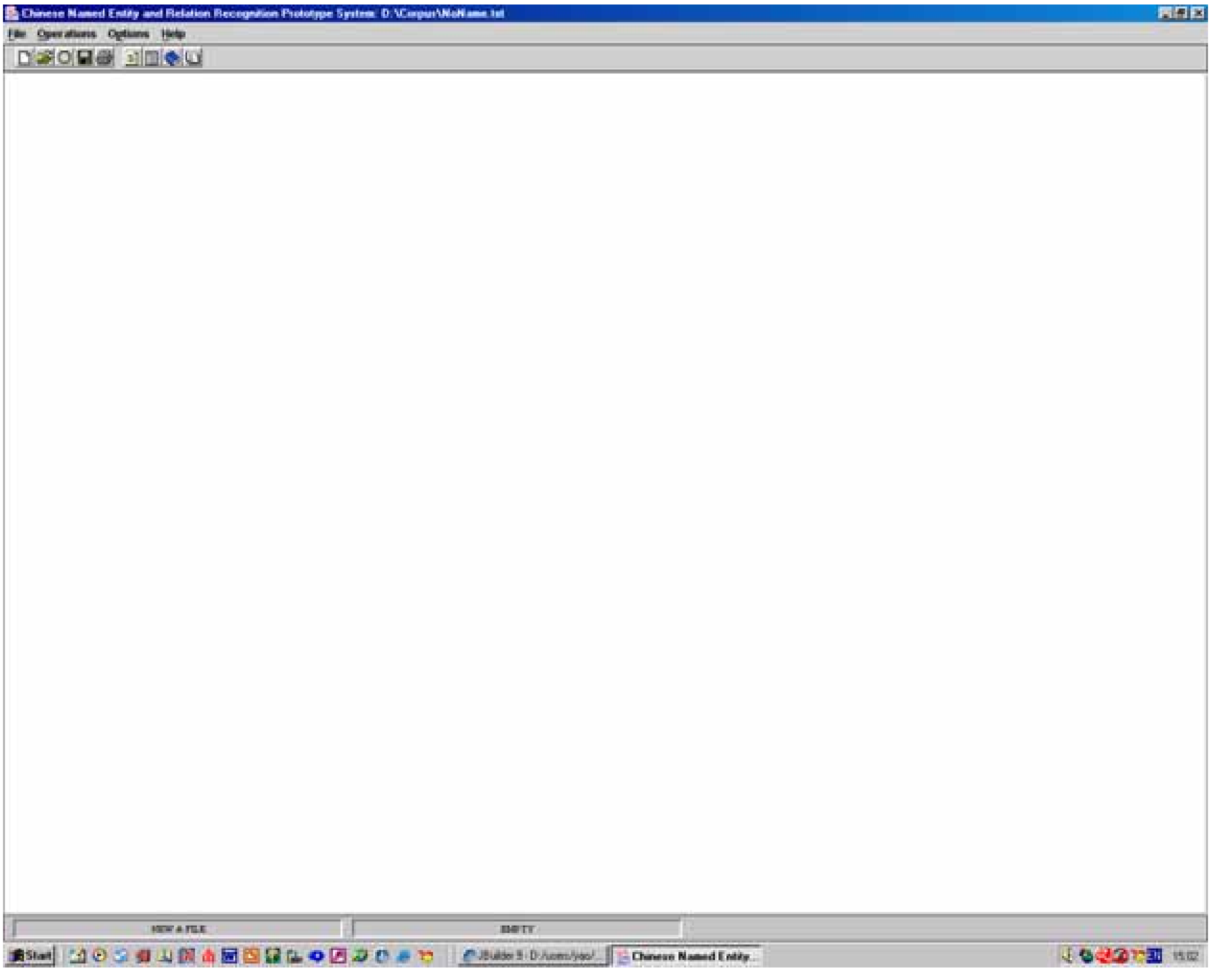
4) GCF and ICF sets can be used for identifying relations and non-relations respectively. For relation identification, we adopt the former feature set because it can capture as many *correct* relations as possible. The feature weight calculation is done only for selected features according to the proportion of their average similarity to the sum of the average similarities of all the selected features.

5) In this example, because three relation instances of CP_TI are very similar, the identification threshold IdenThrh(CP_TI) is *very* high.

Appendix F

System User Interfaces

F.1 Main User Interface (File, Operations, Options, Help)


Chinese Named Entity and Relation Recognition Prototype System: D:\Corpus\NoName.txt
File
Operations
Options
Help
NEW A FILE
EMPTY
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:02

System Functions

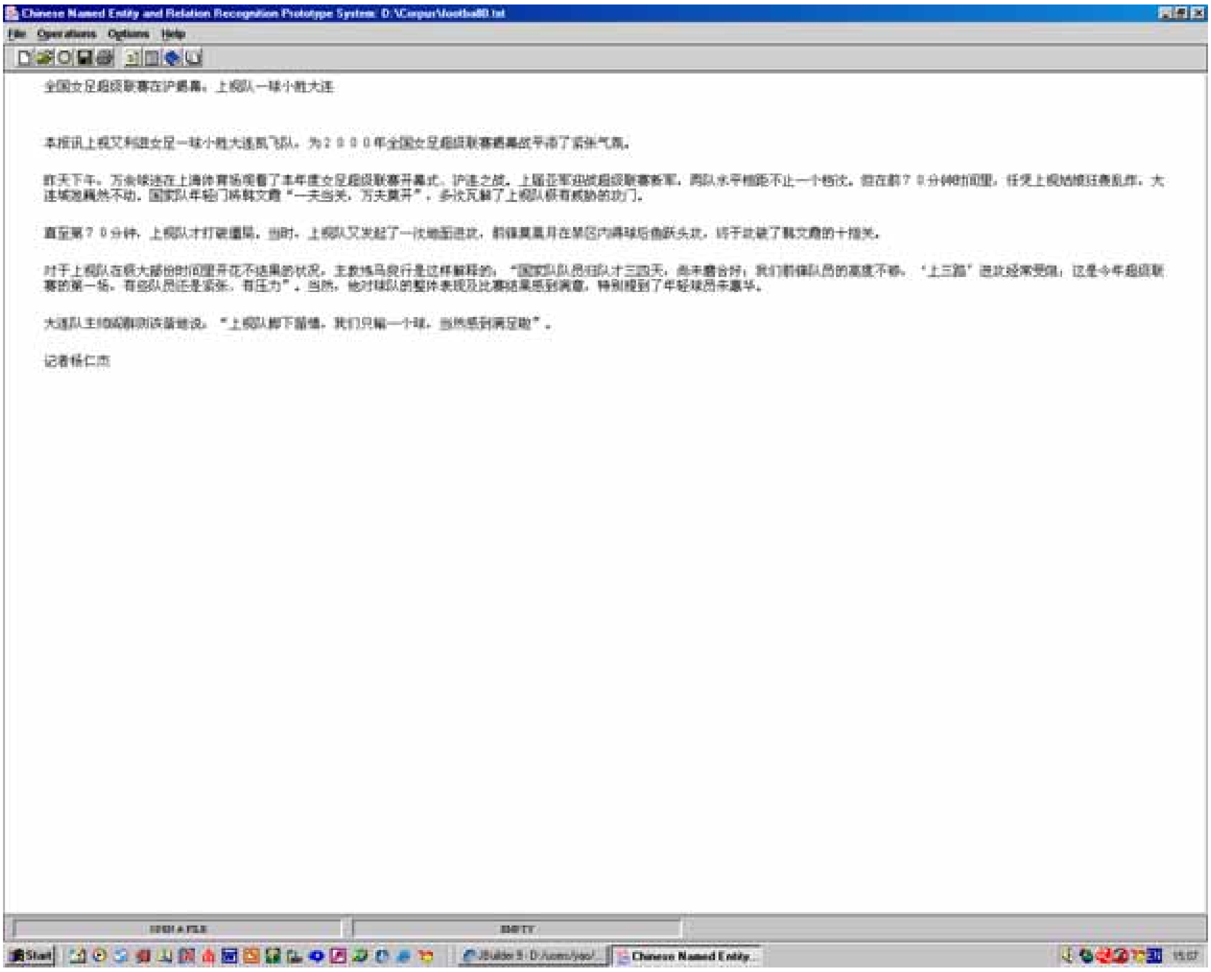
File: New; Open; Download; Close; Save; Save as; Print; Refresh

Operations: Seg. and POS Tag.; Rectify Errors; Named Entity Identification; Named Entity Relation Identification; Display Results; Store Results; Statistics

Options: Choose Font

Help: About Part-of-Speech; About Display Color; About Named Entity Relations; About System

F.2 Open a File from Disk


Chinese Named Entity and Relation Recognition Prototype System: D:\Corpus\football0.txt
File
Operations
Options
Help
OPEN A FILE
EMPTY
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:07

全国女足超级联赛在沪揭幕，上视队一球小胜大连

本报讯上视文利迎女足一球小胜大连凯飞队，为２０００年全国女足超级联赛揭幕战平添了紧张气氛。

昨天下午，万余球迷在上海体育场观看了本年度女足超级联赛开幕式、沪连之战。上届亚军迎战超级联赛冠军，两队水平相距不止一个档次。但在前７０分钟时间里，任凭上视姑娘狂轰乱炸，大连城池巍然不动，国家队年轻门将韩文霞“一夫当关，万夫莫开”，多次瓦解了上视队极有威胁的攻门。

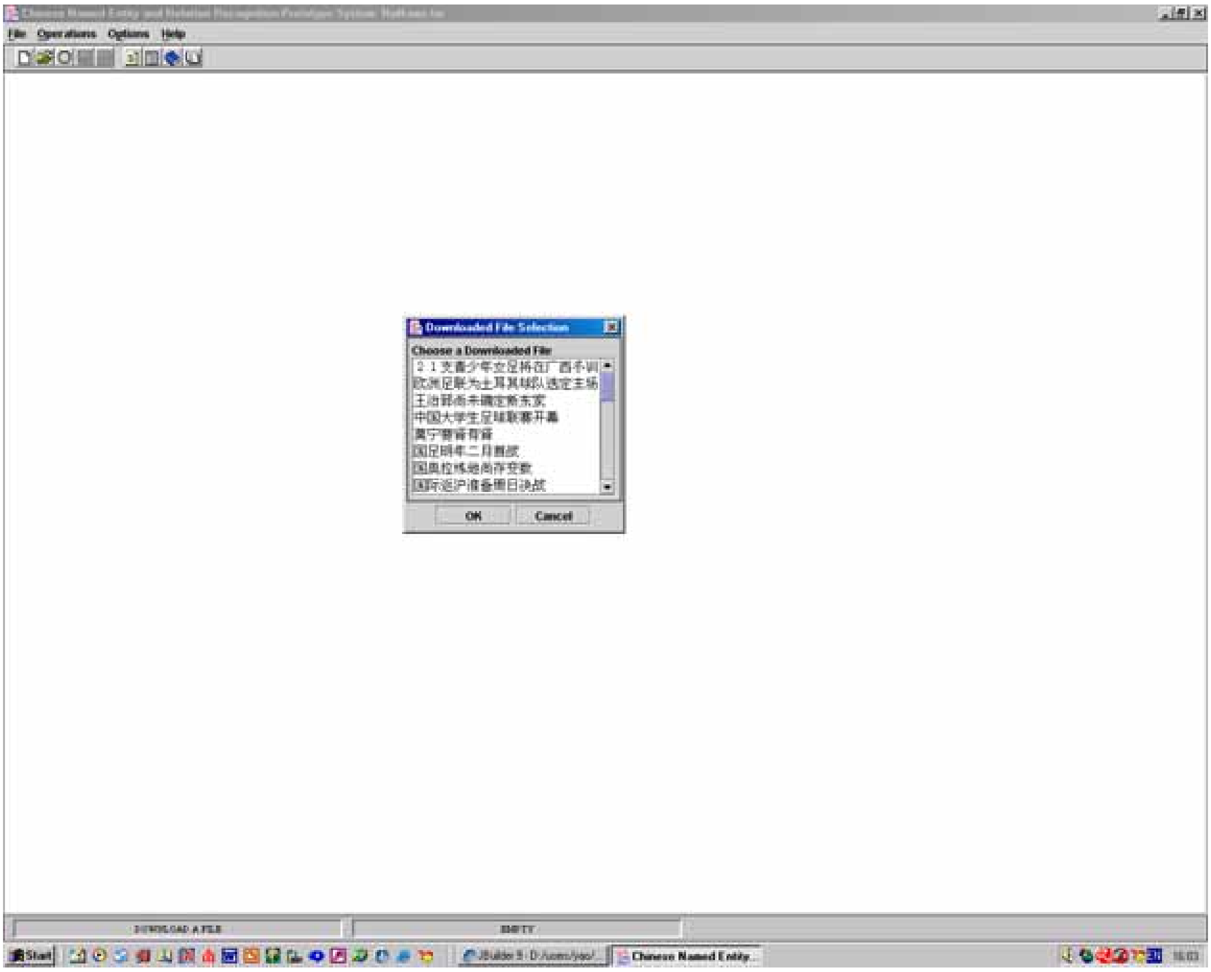
直至第７０分钟，上视队才打破僵局。当时，上视队又发起了一次地面进攻，前锋莫晨月在禁区内摔球后鱼跃头攻，终于攻破了韩文霞的十指关。

对于上视队在很大部份时间里开花不结果的状况，主教练马良行是这样解释的：“国家队队员归队才三四天，尚未磨合好；我们前锋队员的高度不够，‘上三路’进攻经常受阻；这是今年超级联赛的第一场，有些队员还是紧张、有压力”。当然，他对球队的整体表现及比赛结果感到满意，特别提到了年轻球员宋嘉华。

大连队主帅臧淑英赛后说，“上视队脚下细腻，我们只输一个球，当然感到满足”。

记者杨仁杰

F.3 Download a File on Website



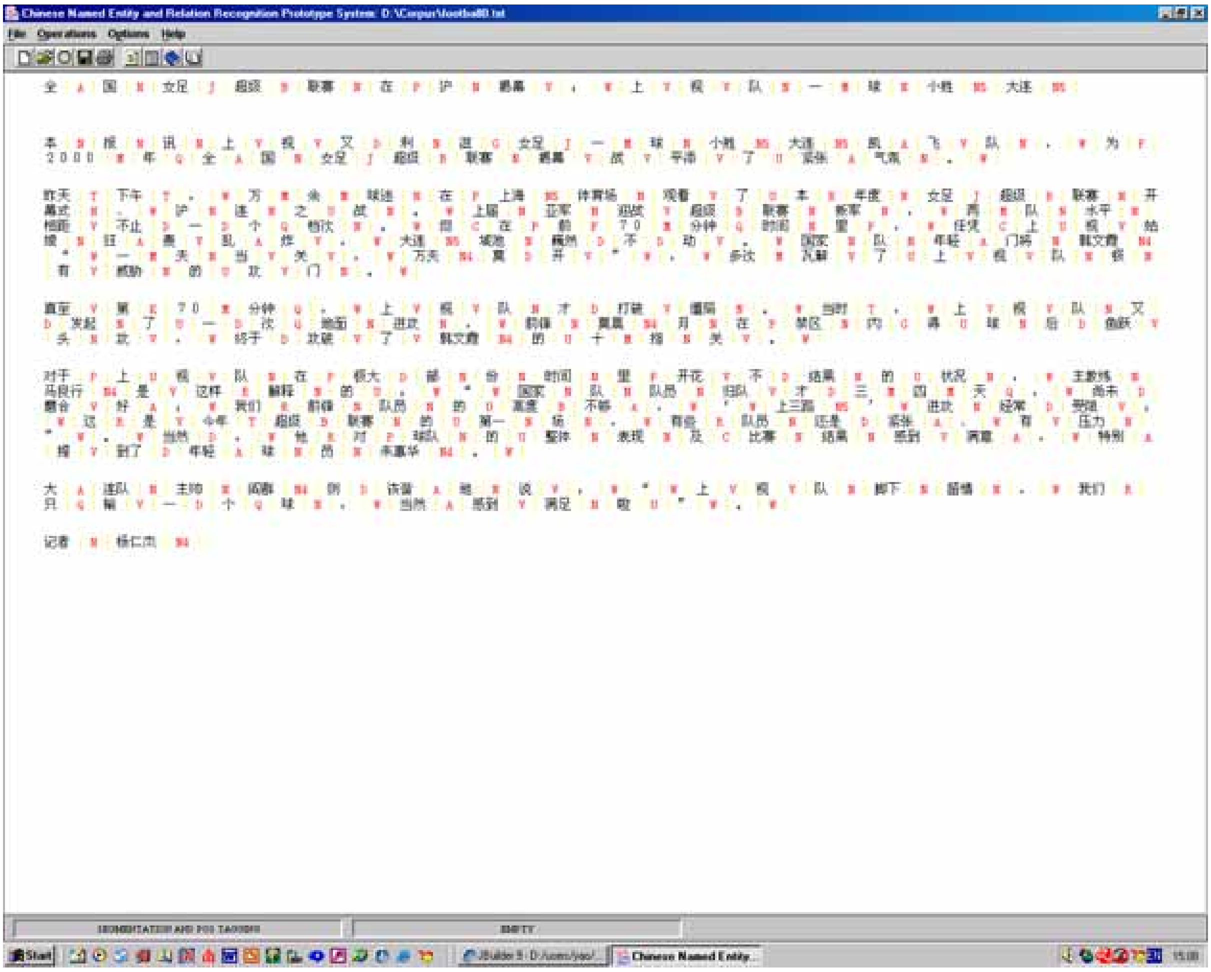
Choose a Downloaded File

２１支青少年女足将在广西集训
欧洲足联为土耳其球队选定主场
王治郅尚未确定新东家
中国大学生足球联赛开幕
冀宁赛背背
国足明年二月首战
国奥拉练地尚存变数
国际巡沪准备周日决战

DOWNLOAD A FILE

EMPTY

F.4 Word Segmentation and Part-of-Speech Tagging


Chinese Named Entity and Relation Recognition Prototype System: D:\Corpus\football0.txt
File
Operations
Options
Help
SEGMENTATION AND POS TAGGING
EMPTY
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:08

全 A 国 N 女足 J 超级 B 联赛 N 在 P 沪 N 揭幕 V ， W 上 V 视 V 队 N 一 M 球 N 小胜 N5 大连 N5

本 R 报 N 讯 N 上 V 视 V 又 D 剩 N 进 G 女足 J 一 M 球 N 小胜 N5 大连 N5 凯 A 飞 V 队 N ， W 为 P 2000 M 年 Q 全 A 国 N 女足 J 超级 B 联赛 N 揭幕 V 战 V 平添 V 了 U 紧张 A 气氛 N 。 W

昨天 T 下午 T ， W 万 N 余 M 球迷 N 在 P 上海 N5 体育场 N 观看 V 了 U 本 R 年度 N 女足 J 超级 B 联赛 N 开幕式 N 、 W 沪 N 连 N 之 U 战 N 。 W 上届 N 亚军 N 迎战 V 超级 B 联赛 N 新军 N ， W 两 M 队 N 水平 N 相距 V 不止 D 一 D 个 Q 档次 N 。 W 但 C 在 P 前 F 70 M 分钟 Q 时间 N 里 F ， W 任凭 C 上 U 视 V 始终 N 狂 A 轰 V 乱 A 炸 V 。 W 大连 N5 城池 N 巍然 D 不 D 动 V 。 W 国家 N 队 N 年轻 A 门将 N 韩文霞 N4 “ W 一 M 夫 N 当 V 关 V 。 W 万夫 N4 莫 D 开 V ” W ， W 多次 M 瓦解 V 了 U 上 V 视 V 队 N 极 N 有 V 威胁 N 的 U 攻 V 门 N 。 W

直至 V 第 K 70 M 分钟 Q ， W 上 V 视 V 队 N 才 D 打破 V 僵局 N 。 W 当时 T ， W 上 V 视 V 队 N 又 D 发起 N 了 U 一 D 次 Q 地面 N 进攻 N ， W 前锋 N 冀晨 N4 月 N 在 P 禁区 N 内 C 得 U 球 N 后 D 鱼跃 V 头 N 攻 V 。 W 终于 D 攻破 V 了 V 韩文霞 N4 的 U 十 M 指 N 关 V 。 W

对于 P 上 U 视 V 队 N 在 P 极大 D 部 N 份 N 时间 N 里 F 开花 V 不 D 结果 N 的 U 状况 N ， W 主教练 N 马良行 N4 是 V 这样 R 解释 N 的 U 。 W “ W 国家 N 队 N 队员 N 归队 V 才 D 三 M 四 M 天 Q ， W 尚未 D 磨合 V 好 A ， W 我们 R 前锋 N 队员 N 的 U 速度 B 不够 A 。 W ‘ W 上三路 N5 ’ W 进攻 N 经常 D 受阻 V 。 W 这 R 是 V 今年 T 超级 B 联赛 N 的 U 第一 M 场 N 。 W 有些 R 队员 N 还是 D 紧张 A 。 W 有 V 压力 N ” W 。 W 当然 D ， W 他 R 对 P 球队 N 的 U 整体 N 表现 N 及 C 比赛 N 结果 N 感到 V 满意 A 。 W 特别 A 提 V 到了 D 年轻 A 球 N 员 N 朱惠华 N4 。 W

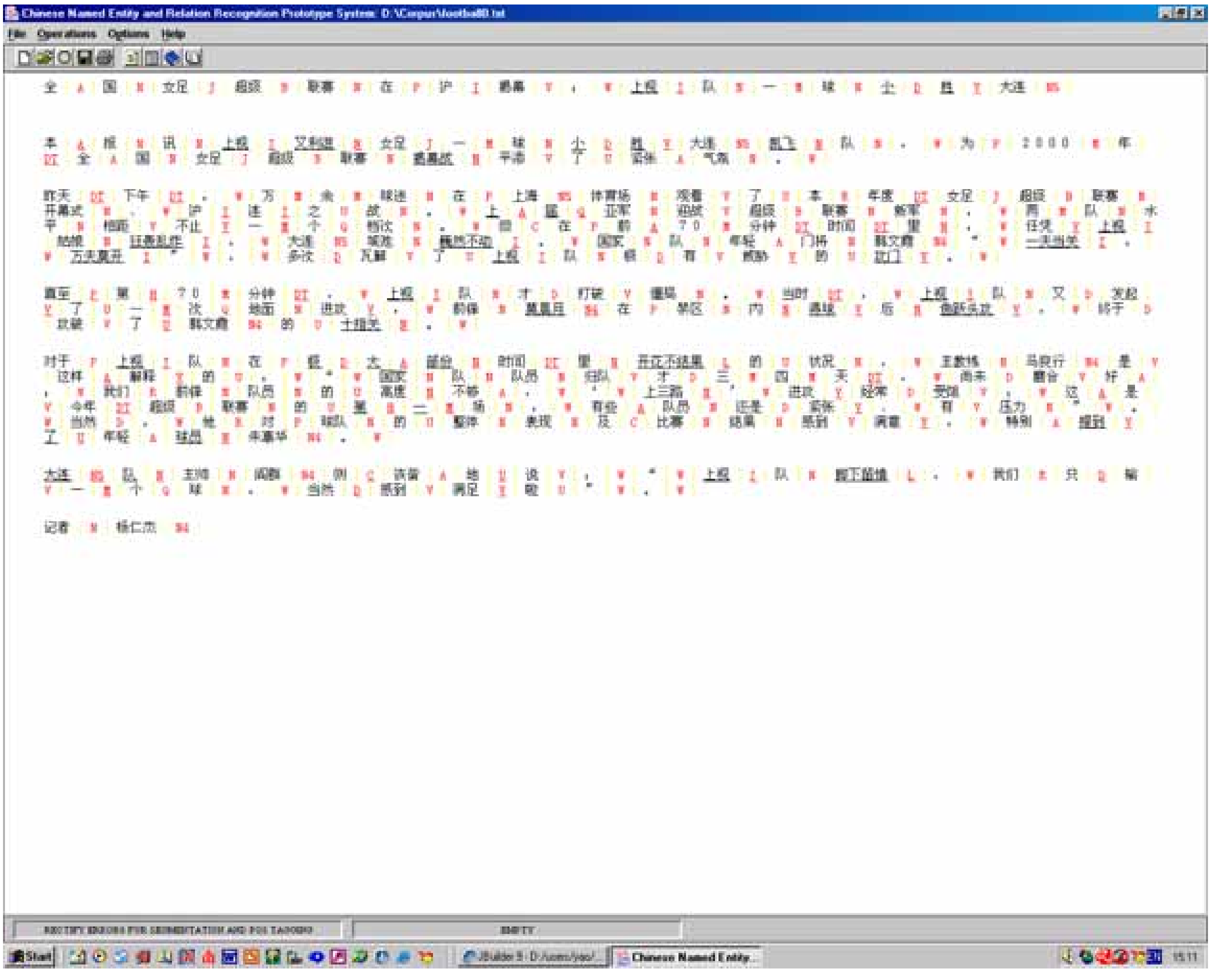
大 A 连队 N 主帅 N 阎鹏 N4 倒 D 诙谐 A 地 U 说 V ， W “ W 上 V 视 V 队 N 脚下 N 留情 N 。 W 我们 R 只 Q 输 V 一 D 个 Q 球 N 。 W 当然 A 感到 V 满足 N 啦 U ” W 。 W

记者 N 杨仁杰 N4

Note

A = Adjective; B = Discrimination; C = Conjunction; D = Adverb; DT = Date or Time; F = Direction; G = Morpheme; H = Prefix; I = Idiom; J = Abbreviated Word; K = Suffix; L = Habitual Word; M = Numeral; N = Common Noun; N1 = Special Noun; N2 = Chinese Personal Surname; N4 = Chinese Personal Name; N5 = Location Name; N7 = Transliterated Name; P = Proposition; Q = Quantifier; R = Pronoun; U = Auxiliary Word; V = Verb; W = Punctuation Mark;

F.5 Automatically Repair Errors for Segmentation and Part-of-Speech Tagging (See the Underlined Part in the Text)


Chinese Named Entity and Relation Recognition Prototype System: D:\Corpus\football0.txt
File
Operations
Options
Help
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:11

全 国 女足 超级 联赛 在 沪 揭幕 ， 上视 队 一 球 小 胜 大连

本 报 讯 上视 艾利建 女足 一 球 小 胜 大连 凯飞 队 。 为 2000 年 全 国 女足 超级 联赛 揭幕战 平添 了 紧张 气氛 。

昨天 下午 ， 万 余 球迷 在 上海 体育场 观看 了 本 年度 女足 超级 联赛 开幕式 。 沪 连 之 战 。 上 届 亚军 迎战 超级 联赛 新军 ， 两 队 水平 相距 不止 一 个 档次 。 但 在 前 70 分钟 时间 里 ， 任凭 上视 姑娘 狂轰乱炸 ， 大连 城池 巍然不动 ， 国家 队 年轻 门将 韩文霞 " 一夫当关 ， 万夫莫开 " ， 多次 瓦解 了 上视 队 极 有 威胁 的 攻门 。

直至 第 70 分钟 ， 上视 队 才 打破 僵局 。 当时 ， 上视 队 又 发起 了 一 次 地面 进攻 ， 前锋 [illegible] 在 禁区 内 晃过 [illegible] 后 [illegible] ， 终于 攻破 了 韩文霞 的 [illegible] 。

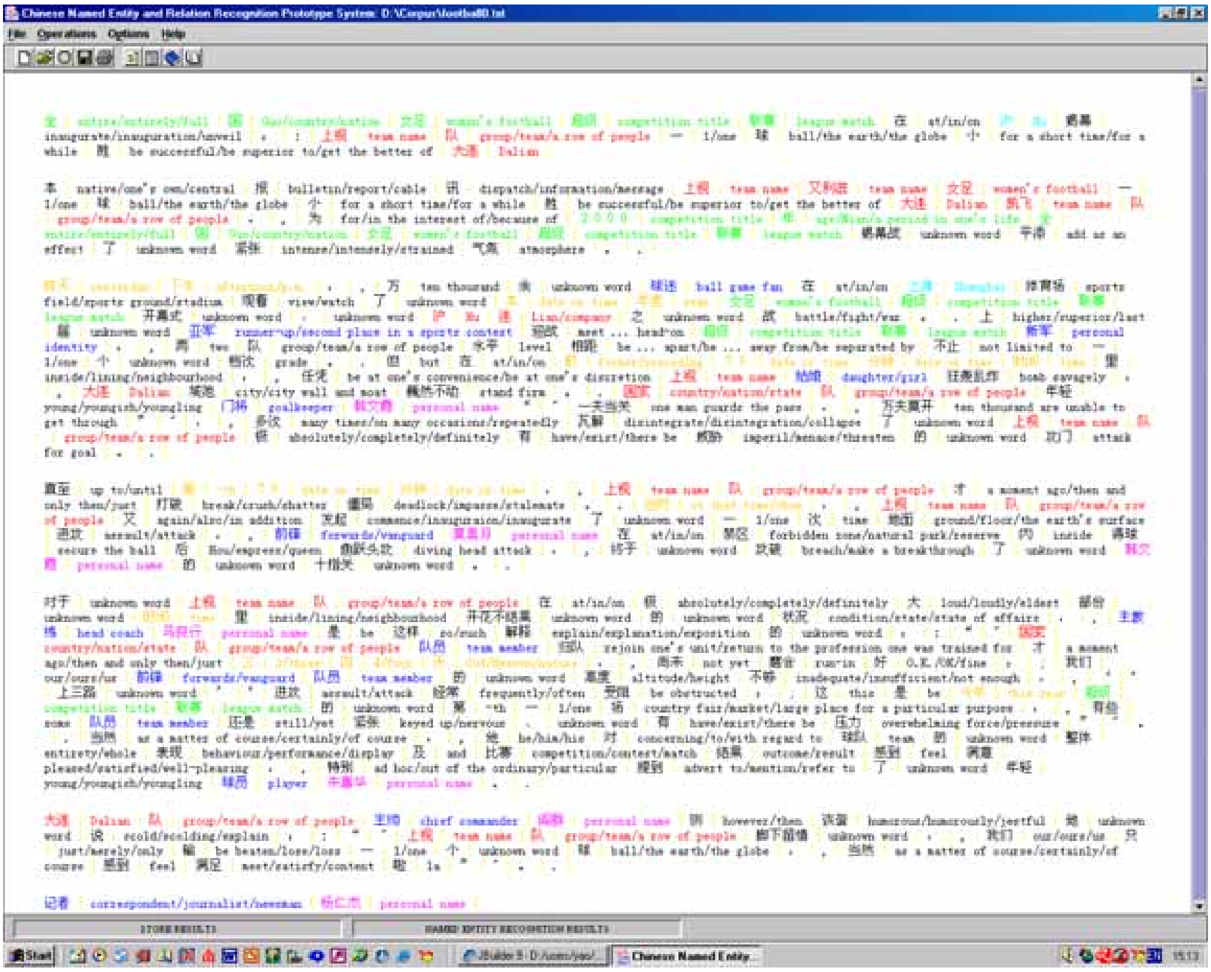
对于 上视 队 在 很 大 部分 时间 里 开花不结果 的 状况 ， 主教练 马良行 是 这样 解释 的 ： " 国家 队 队员 归队 才 三 四 天 ， 尚未 磨合 好 ， 我们 前锋 队员 的 高度 不够 ， 上三路 进攻 经常 受阻 。 这 是 今年 超级 联赛 的 第 一 场 ， 有些 队员 还是 紧张 ， 有 压力 " 。 当然 ， 他 对 球队 的 整体 表现 及 比赛 结果 感到 满意 ， 特别 提到 了 年轻 球员 朱赛华 。

大连 队 主帅 阎鲁 则 诙谐 地 说 ： " 上视 队 脚下留情 。 我们 只 输 一 个 球 ， 当然 感到 满足 啦 " 。

记者 杨仁杰

RECTIFY ERRORS FOR SEGMENTATION AND POS TAGGING | EMPTY

F.6 Identification Result 1 for Named Entities (Chinese-English Representation)


Chinese Named Entity and Relation Recognition Prototype System: D:\Corpus\football0.txt
File Operations Options Help
STORE RESULTS
NAMED ENTITY RECOGNITION RESULTS
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:13

全 entire/entirely/full 国 Guo/country/nation 女足 women's football 超级 competition title 联赛 league match 在 at/in/on 沪 Hu 揭幕 inaugurate/inauguration/unveil ， ： 上视 team name 队 group/team/a row of people 一 1/one 球 ball/the earth/the globe 小 for a short time/for a while 胜 be successful/be superior to/get the better of 大连 Dalian

本 native/one's own/central 报 bulletin/report/cable 讯 dispatch/information/message 上视 team name 又利胜 team name 女足 women's football 一 1/one 球 ball/the earth/the globe 小 for a short time/for a while 胜 be successful/be superior to/get the better of 大连 Dalian 凯飞 team name 队 group/team/a row of people ， 为 for/in the interest of/because of ２０００ competition title 年 age/Nian/a period in one's life 全 entire/entirely/full 国 Guo/country/nation 女足 women's football 超级 competition title 联赛 league match 揭幕战 unknown word 平添 add as an effect 了 unknown word 紧张 intense/intensely/strained 气氛 atmosphere 。

昨天 yesterday 下午 afternoon/p.m. ， 万 ten thousand 余 unknown word 球迷 ball game fan 在 at/in/on 上海 Shanghai 体育场 sports field/sports ground/stadium 观看 view/watch 了 unknown word 本 date or time 年度 year 女足 women's football 超级 competition title 联赛 league match 开幕式 unknown word · unknown word 沪 Hu 连 Lian/company 之 unknown word 战 battle/fight/war 。 上 higher/superior/last 届 unknown word 亚军 runner-up/second place in a sports contest 相逢 meet ... head-on 超级 competition title 联赛 league match 新军 personal identity ， 两 two 队 group/team/a row of people 水平 level 相距 be ... apart/be ... away from/be separated by 不止 not limited to 一 1/one 个 unknown word 档次 grade 。 但 but 在 at/in/on 前 former/preceding ７７ date or time 分钟 date or time 时间 time 里 inside/lining/neighbourhood ， 任凭 be at one's convenience/be at one's discretion 上视 team name 姑娘 daughter/girl 狂轰乱炸 bomb savagely ， 大连 Dalian 城池 city/city wall and moat 巍然不动 stand firm 。 国家 country/nation/state 队 group/team/a row of people 年轻 young/youngish/youngling 门将 goalkeeper 韩文霞 personal name " " 一夫当关 one man guards the pass ， 万夫莫开 ten thousand are unable to get through " " 。 多次 many times/on many occasions/repeatedly 瓦解 disintegrate/disintegration/collapse 了 unknown word 上视 team name 队 group/team/a row of people 极 absolutely/completely/definitely 有 have/exist/there be 威胁 imperil/menace/threaten 的 unknown word 攻门 attack for goal 。

直至 up to/until 第 -th ７８ date or time 分钟 date or time ， 上视 team name 队 group/team/a row of people 才 a moment ago/then and only then/just 打破 break/crush/shatter 僵局 deadlock/impasse/stalemate 。 此时 at that time/then ， 上视 team name 队 group/team/a row of people 又 again/also/in addition 发起 commence/inauguraion/inaugurate 了 unknown word 一 1/one 次 time 地面 ground/floor/the earth's surface 进攻 assault/attack ， 前锋 forwards/vanguard 冀秀月 personal name 在 at/in/on 禁区 forbidden zone/natural park/reserve 内 inside 得球 secure the ball 后 Hou/empress/queen 鱼跃头攻 diving head attack ， 将球 unknown word 攻破 breach/make a breakthrough 了 unknown word 韩文霞 personal name 的 unknown word 十指关 unknown word 。

对于 unknown word 上视 team name 队 group/team/a row of people 在 at/in/on 很 absolutely/completely/definitely 大 loud/loudly/eldest 部份 unknown word 时间 time 里 inside/lining/neighbourhood 开花不结果 unknown word 的 unknown word 状况 condition/state/state of affairs ， 主教练 head coach 马良行 personal name 是 be 这样 so/such 解释 explain/explanation/exposition 的 unknown word ： " 国家 country/nation/state 队 group/team/a row of people 队员 team member 归队 rejoin one's unit/return to the profession one was trained for 才 a moment ago/then and only then/just 三 3/three 四 4/four 天 Day/Heaven/nature ， 尚未 not yet 磨合 run-in 好 O.K./OK/fine ； 我们 our/ours/us 前锋 forwards/vanguard 队员 team member 的 unknown word 高度 altitude/height 不够 inadequate/insufficient/not enough ， ' ' 上三路 unknown word ' ' 进攻 assault/attack 经常 frequently/often 受阻 be obstructed 。 这 this 是 be 今年 this year 超级 competition title 联赛 league match 的 unknown word 第 -th 一 1/one 场 country fair/market/large place for a particular purpose ， 有些 some 队员 team member 还是 still/yet 紧张 keyed up/nervous 。 unknown word 有 have/exist/there be 压力 overwhelming force/pressure " 。 当然 as a matter of course/certainly/of course ， 他 he/him/his 对 concerning/to/with regard to 球队 team 的 unknown word 整体 entirety/whole 表现 behaviour/performance/display 及 and 比赛 competition/contest/match 结果 outcome/result 感到 feel 满意 pleased/satisfied/well-pleasing ， 特别 ad hoc/out of the ordinary/particular 提到 advert to/mention/refer to 了 unknown word 年轻 young/youngish/youngling 球员 player 朱惠华 personal name 。

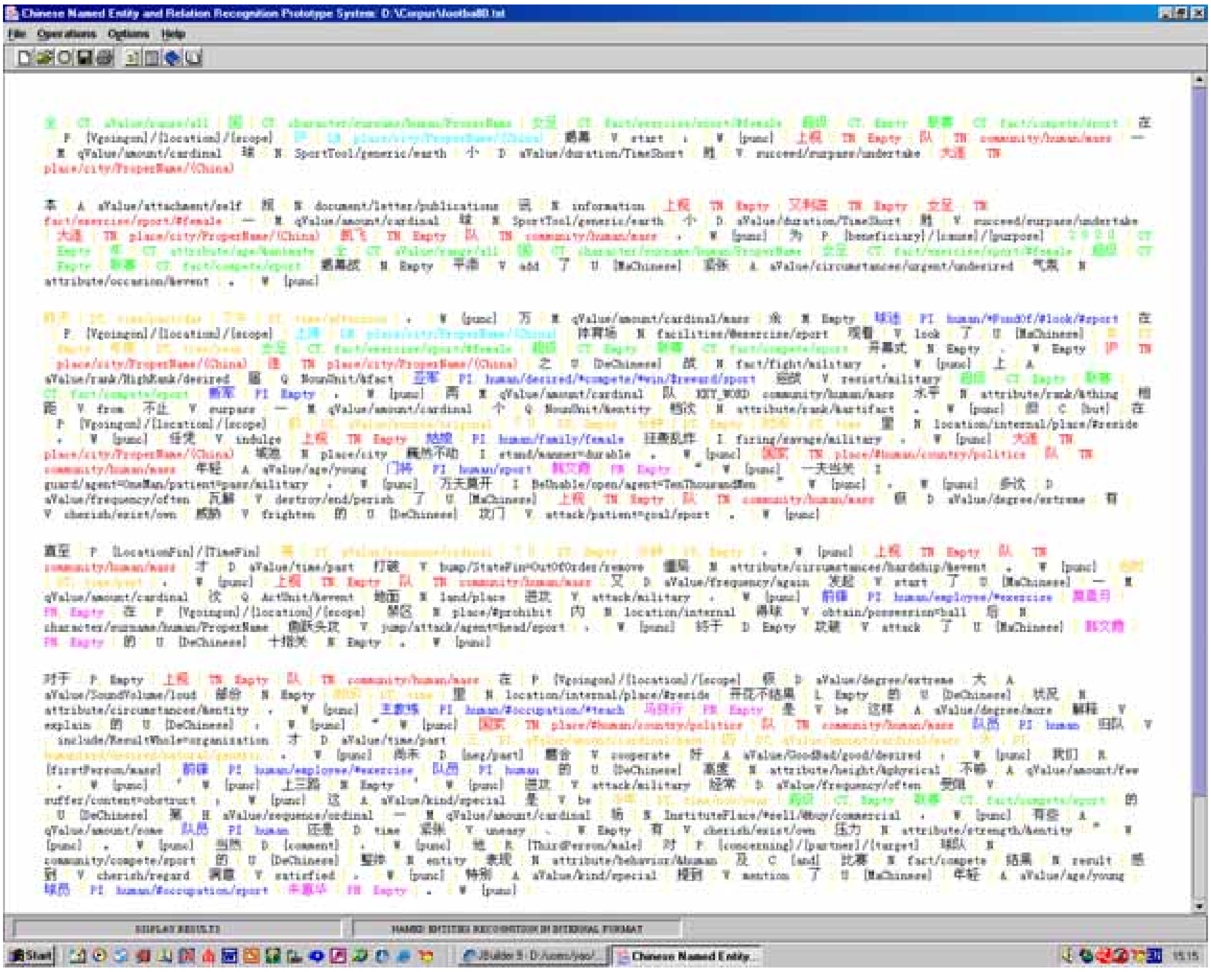
大连 Dalian 队 group/team/a row of people 主帅 chief commander 俞鹏 personal name 则 however/then 诙谐 humorous/humorously/jestful 地 unknown word 说 scold/scolding/explain ： " 上视 team name 队 group/team/a row of people 脚下留情 unknown word ， 我们 our/ours/us 只 just/merely/only 输 be beaten/lose/loss 一 1/one 个 unknown word 球 ball/the earth/the globe ， 当然 as a matter of course/certainly/of course 感到 feel 满足 meet/satisfy/content 啦 la " 。

记者 correspondent/journalist/newsman 杨仁杰 personal name

Note

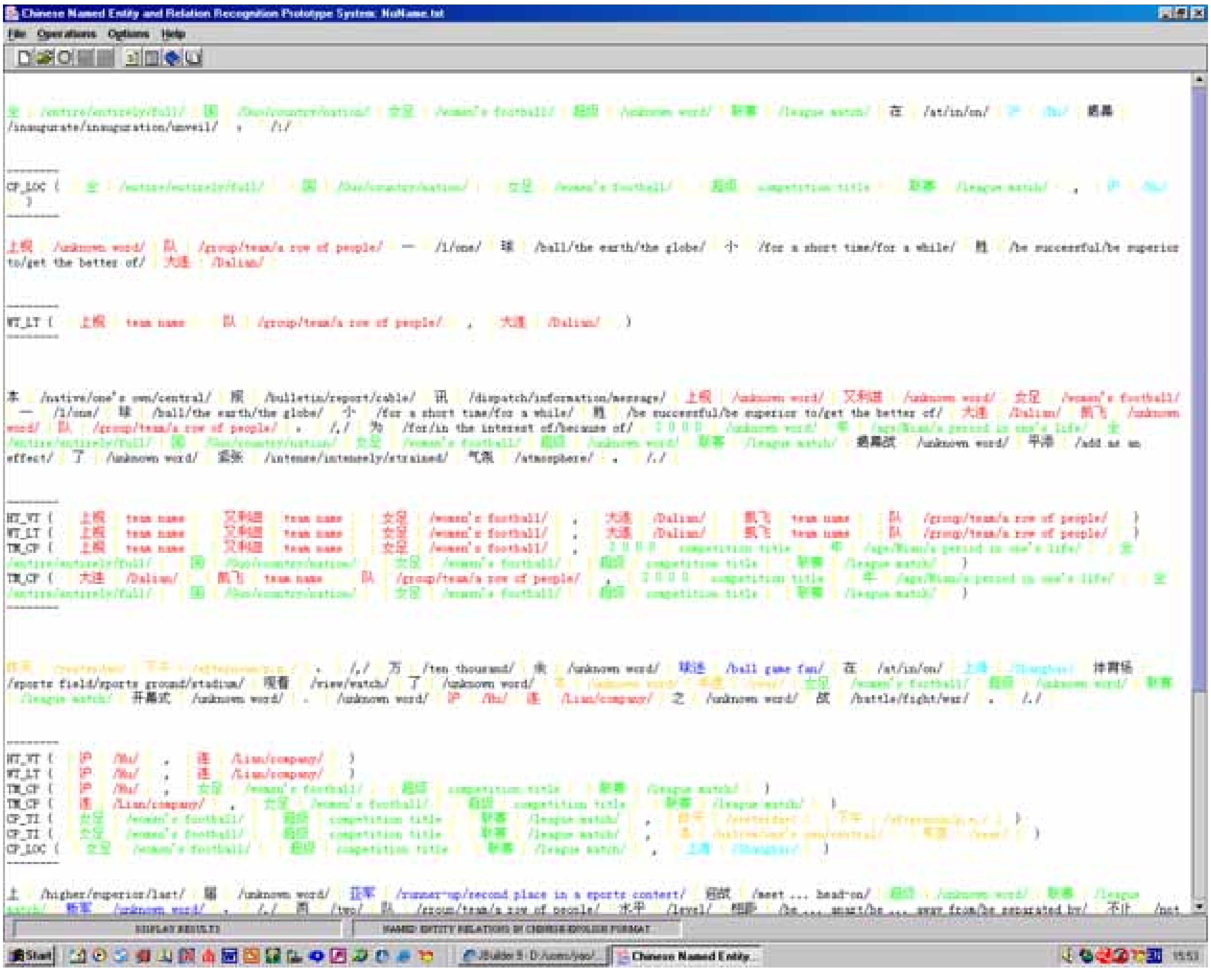
Black = General Word and Punctuation; Yellow = Separator; Red = Team Name; Green = Competition Title; Blue - Personal Identity; Magenta = Personal Name; Orange = Date or Time; Cyan = Location Name

F.7 Identification Result 2 for Named Entities (Internal Information Representation)


Chinese Named Entity and Relation Recognition Prototype System: D:\Corpus\football0.txt
File
Operations
Options
Help
DISPLAY RESULTS
NAMED ENTITIES RECOGNITION IN INTERNAL FORMAT
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:15

```
全 CT aValue/range/all 国 CT character/surname/human/ProperName 女足 CT fact/exercise/sport/#female 超级 CT Empty 联赛 CT fact/compete/sport 在
 P [Vgoingon]/[location]/[scope] 沪 LN place/city/ProperName/(China) 揭幕 V start ， W [punc] 上视 TN Empty 队 TN community/human/mass 一
 M qValue/amount/cardinal 球 N SportTool/generic/earth 小 D aValue/duration/TimeShort 胜 V succeed/surpass/undertake 大连 TN
place/city/ProperName/(China)

本 A aValue/attachment/self 报 N document/letter/publications 讯 N information 上视 TN Empty 又利国 TN Empty 女足 TN
fact/exercise/sport/#female 一 M qValue/amount/cardinal 球 N SportTool/generic/earth 小 D aValue/duration/TimeShort 胜 V succeed/surpass/undertake
 大连 TN place/city/ProperName/(China) 凯飞 TN Empty 队 TN community/human/mass ， W [punc] 为 P [beneficiary]/[cause]/[purpose] 2000 CT
 Empty 年 CT attribute/age/#animate 全 CT aValue/range/all 国 CT character/surname/human/ProperName 女足 CT fact/exercise/sport/#female 超级 CT
 Empty 联赛 CT fact/compete/sport 揭幕战 N Empty 平添 V add 了 U [MaChinese] 紧张 A aValue/circumstances/urgent/undesired 气氛 N
attribute/occasion/#event 。 W [punc]

昨天 DT time/yesterday 下午 DT time/afternoon ， W [punc] 万 M qValue/amount/cardinal/mass 余 M Empty 球迷 PI human/*FondOf/#look/#sport 在
 P [Vgoingon]/[location]/[scope] 上海 LN place/city/ProperName/(China) 体育场 N facilities/#exercise/sport 观看 V look 了 U [MaChinese] 本 DT
 Empty 年度 DT time/year 女足 CT fact/exercise/sport/#female 超级 CT Empty 联赛 CT fact/compete/sport 开幕式 N Empty 。 W Empty 沪 TN
 place/city/ProperName/(China) 连 TN place/city/ProperName/(China) 之 U [DeChinese] 战 N fact/fight/military ， W [punc] 上 A
aValue/rank/HighRank/desired 届 Q NounUnit/#fact 亚军 PI human/desired/*compete/*win/$reward/sport 迎战 V resist/military 超级 CT Empty 联赛
CT fact/compete/sport 冠军 PI Empty ， W [punc] 两 M qValue/amount/cardinal 队 KEY_WORD community/human/mass 水平 N attribute/rank/&thing 相
距 V from 不止 V surpass 一 M qValue/amount/cardinal 个 Q NounUnit/&entity 档次 N attribute/rank/&artifact 。 W [punc] 但 C [but] 在
 P [Vgoingon]/[location]/[scope] 前 DT aValue/sequence/original 70 DT Empty 分钟 DT Empty 时间 DT time 里 N location/internal/place/#reside
 ， W [punc] 任凭 V indulge 上视 TN Empty 姑娘 PI human/family/female 狂轰乱炸 I firing/savage/military ， W [punc] 大连 TN
place/city/ProperName/(China) 城池 N place/city 巍然不动 I stand/manner=durable ， W [punc] 国家 TN place/#human/country/politics 队 TN
community/human/mass 年轻 A aValue/age/young 门将 PI human/sport 赵文莉 FN Empty " W [punc] 一夫当关 I
guard/agent=OneMan/patient=pass/military ， W [punc] 万夫莫开 I BeUnable/open/agent=TenThousandMen " W [punc] 。 W [punc] 多次 D
aValue/frequency/often 瓦解 V destroy/end/perish 了 U [MaChinese] 上视 TN Empty 队 TN community/human/mass 极 D aValue/degree/extreme 有
V cherish/exist/own 威胁 V frighten 的 U [DeChinese] 攻门 V attack/patient=goal/sport 。 W [punc]

直至 P [LocationFin]/[TimeFin] 第 DT qValue/sequence/ordinal 70 DT Empty 分钟 DT Empty ， W [punc] 上视 TN Empty 队 TN
community/human/mass 才 D aValue/time/past 打破 V bump/StateFin=OutOfOrder/remove 僵局 N attribute/circumstances/hardship/#event ， W [punc] 此时
 DT time/past ， W [punc] 上视 TN Empty 队 TN community/human/mass 又 D aValue/frequency/again 发起 V start 了 U [MaChinese] 一 M
qValue/amount/cardinal 次 Q ActUnit/&event 地面 N land/place 进攻 V attack/military ， W [punc] 前锋 PI human/employee/*exercise 戴盛月
FN Empty 在 P [Vgoingon]/[location]/[scope] 禁区 N place/#prohibit 内 N location/internal 得球 V obtain/possession=ball 后 N
character/surname/human/ProperName 鱼跃头攻 V jump/attack/agent=head/sport ， W [punc] 终于 D Empty 攻破 V attack 了 U [MaChinese] 赵文莉
FN Empty 的 U [DeChinese] 十指关 N Empty 。 W [punc]

对于 P Empty 上视 TN Empty 队 TN community/human/mass 在 P [Vgoingon]/[location]/[scope] 极 D aValue/degree/extreme 大 A
aValue/SoundVolume/loud 部份 N Empty 时间 DT time 里 N location/internal/place/#reside 开花不结果 L Empty 的 U [DeChinese] 状况 N
attribute/circumstances/&entity ， W [punc] 王教练 PI human/#occupation/*teach 马良行 FN Empty 是 V be 这样 A aValue/degree/more 解释 V
explain 的 U [DeChinese] ： W [punc] " W [punc] 国家 TN place/#human/country/politics 队 TN community/human/mass 队员 PI human 归队 V
 include/ResultWhole=organization 才 D aValue/time/past 三 DT aValue/amount/cardinal/mass 四 DT aValue/amount/cardinal/mass 天 DT
humanized/desired/natural/growth ， W [punc] 尚未 D [neg/past] 磨合 V cooperate 好 A aValue/GoodBad/good/desired ， W [punc] 我们 R
[firstPerson/mass] 前锋 PI human/employee/*exercise 队员 PI human 的 U [DeChinese] 高度 N attribute/height/&physical 不够 A qValue/amount/few
 ， W [punc] ' W [punc] 上三路 N Empty ' W [punc] 进攻 V attack/military 经常 D aValue/frequency/often 受阻 V
suffer/content=obstruct ， W [punc] 这 A aValue/kind/special 是 V be 今年 DT time/now/year 超级 CT Empty 联赛 CT fact/compete/sport 的
 U [DeChinese] 第 M aValue/sequence/ordinal 一 M qValue/amount/cardinal 场 N InstituteFlace/#sell/#buy/commercial ， W [punc] 有些 A
qValue/amount/some 队员 PI human 还是 D time 紧张 V uneasy ， W Empty 有 V cherish/exist/own 压力 N attribute/strength/&entity " W
[punc] 。 W [punc] 当然 D [comment] ， W [punc] 他 R [ThirdPerson/male] 对 P [concerning]/[partner]/[target] 球队 N
community/compete/sport 的 U [DeChinese] 整体 N entity 表现 N attribute/behavior/&human 及 C [and] 比赛 N fact/compete 结果 N result 感
到 V cherish/regard 满意 V satisfied 。 W [punc] 特别 A aValue/kind/special 提到 V mention 了 U [MaChinese] 年轻 A aValue/age/young
球员 PI human/#occupation/sport 朱暖华 FN Empty 。 W [punc]
```

F.8 Identification Result for Named Entity Relations


Chinese Named Entity and Relation Recognition Prototype System: NoName.txt
File Operations Options Help
DISPLAY RESULTS
NAMED ENTITY RELATIONS IN CHINESE-ENGLISH FORMAT
Start
JBuilder 5 - D:/users/yao/...
Chinese Named Entity...
15:53

```
全 /entire/entirely/full/ 国 /Guo/country/nation/ 女足 /women's football/ 超级 /unknown word/ 联赛 /league match/ 在 /at/in/on/ 沪 /Hu/ 揭幕
/inaugurate/inauguration/unveil/ 。 /./

----------
CP_LOC ( 全 /entire/entirely/full/ 国 /Guo/country/nation/ 女足 /women's football/ 超级 competition title 联赛 /league match/ , 沪 /Hu/
 )
----------

上视 /unknown word/ 队 /group/team/a row of people/ 一 /1/one/ 球 /ball/the earth/the globe/ 小 /for a short time/for a while/ 胜 /be successful/be superior
to/get the better of/ 大连 /Dalian/

----------
WT_LT ( 上视 team name 队 /group/team/a row of people/ , 大连 /Dalian/ )
----------

本 /native/one's own/central/ 报 /bulletin/report/cable/ 讯 /dispatch/information/message/ 上视 /unknown word/ 艾利亚 /unknown word/ 女足 /women's football/
 一 /1/one/ 球 /ball/the earth/the globe/ 小 /for a short time/for a while/ 胜 /be successful/be superior to/get the better of/ 大连 /Dalian/ 凯飞 /unknown
word/ 队 /group/team/a row of people/ , /,/ 为 /for/in the interest of/because of/ 2000 /unknown word/ 年 /age/Nian/a period in one's life/ 全
/entire/entirely/full/ 国 /Guo/country/nation/ 女足 /women's football/ 超级 /unknown word/ 联赛 /league match/ 揭幕战 /unknown word/ 平添 /add as an
effect/ 了 /unknown word/ 紧张 /intense/intensely/strained/ 气氛 /atmosphere/ 。 /./

----------
HT_VT ( 上视 team name 艾利亚 team name 女足 /women's football/ ; 大连 /Dalian/ 凯飞 team name 队 /group/team/a row of people/ )
WT_LT ( 上视 team name 艾利亚 team name 女足 /women's football/ ; 大连 /Dalian/ 凯飞 team name 队 /group/team/a row of people/ )
TM_CP ( 上视 team name 艾利亚 team name 女足 /women's football/ ; 2000 competition title 年 /age/Nian/a period in one's life/ 全
/entire/entirely/full/ 国 /Guo/country/nation/ 女足 /women's football/ 超级 competition title 联赛 /league match/ )
TM_CP ( 大连 /Dalian/ 凯飞 team name 队 /group/team/a row of people/ , 2000 competition title 年 /age/Nian/a period in one's life/ 全
/entire/entirely/full/ 国 /Guo/country/nation/ 女足 /women's football/ 超级 competition title 联赛 /league match/ )
----------

昨天 /yesterday/ 下午 /afternoon/p.m./ , /,/ 万 /ten thousand/ 余 /unknown word/ 球迷 /ball game fan/ 在 /at/in/on/ 上海 /Shanghai/ 体育场
/sports field/sports ground/stadium/ 观看 /view/watch/ 了 /unknown word/ 本 /unknown word/ 年度 /year/ 女足 /women's football/ 超级 /unknown word/ 联赛
/league match/ 开幕式 /unknown word/ , /unknown word/ 沪 /Hu/ 连 /Lian/company/ 之 /unknown word/ 战 /battle/fight/war/ 。 /./

----------
HT_VT ( 沪 /Hu/ ; 连 /Lian/company/ )
WT_LT ( 沪 /Hu/ ; 连 /Lian/company/ )
TM_CP ( 沪 /Hu/ ; 女足 /women's football/ 超级 competition title 联赛 /league match/ )
TM_CP ( 连 /Lian/company/ , 女足 /women's football/ 超级 competition title 联赛 /league match/ )
CP_TI ( 女足 /women's football/ 超级 competition title 联赛 /league match/ , 昨天 /yesterday/ 下午 /afternoon/p.m./ )
CP_TI ( 女足 /women's football/ 超级 competition title 联赛 /league match/ , 本 /native/one's own/central/ 年度 /year/ )
CP_LOC ( 女足 /women's football/ 超级 competition title 联赛 /league match/ , 上海 /Shanghai/ )
----------

上 /higher/superior/last/ 届 /unknown word/ 亚军 /runner-up/second place in a sports contest/ 相遇 /meet ... head-on/ 超级 /unknown word/ 联赛 /league
match/ 新军 /unknown word/ , /,/ 两 /two/ 队 /group/team/a row of people/ 水平 /level/ 相距 /be ... apart/be ... away from/be separated by/ 不止 /not
```

Note

PS_TM = Person - Team; PS_CP = Person - Competition; PS_CPC = Person - City / Province / Country; PS_ID = Person - Identification; HT_VT = Home Team - Visiting Team; WT_LT = Winning Team - Losing Team; DT_DT = Draw Team - Draw Team; TM_CP = Team - Competition; TM_CPC = Team - City / Province / Country; ID_TM = Identification - Team; CP_DA = Competition - Date; CP_TI = Competition - Time; CP_LOC = Competition - Location; LOC_ CPC = Location - City / Province / Country

Bibliography

S. Abney. 1990. *Rapid Incremental Parsing with Repair*. In the Proc. of the 6th New OED Conference: Electronic Text Research, pages 1-9. University of Waterloo, Ontario, Canada.

S. Abney. 1996. *Partial Parsing via Finite-State Cascades*. In Proceedings of the ESSLLI '96 Robust Parsing Workshop. Prague, Czech Republic.

S. Abney. 1997. *Part-of-Speech Tagging and Partial Parsing*. In S. Young and G. Bloothooft, editors, Corpus-Based Methods in Language and Speech Processing, pages 118-136. Kluwer Academic Publishers, Dordrecht.

D. Aha, D. Kibler, and M. Albert. 1991. *Instance-Based Learning Algorithms*. Machine Learning, Vol. 6, No.1, pages 37-66.

A. Aho, J. Hopcroft, and J. Ullman. 1983. *Data Structures and Algorithms*. Addison-Wesley Publishing Company. Reading, Massachusetts, USA.

D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, and M. Tyson. 1995. *SRI International FASTUS System MUC-6 Test Results and Analysis*. In Proc. of the Sixth Message Understanding Conference (MUC-6), Columbia, Maryland. NIST. Morgen-Kaufmann Publishers.

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. 1993. *FASTUS: A Finite-State Processor for Information Extraction from Real-World Text*. In Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93), pages 1172-1178. Chambery, France.

D. Appelt and D. Israel. 1999. *Introduction to Information Extraction Technology*. IJCAI-99 Tutorial. Stockholm, Sweden. http://www.ai.sri.com/~appelt/ie-tutorial/.

C. Baker, C. Fillmore, and J. Lowe. 1998. *The Berkeley FrameNet Project*. In Proc. of COLING-ACL '98, pages 86-90. Montreal, Canada.

L. Baum, T. Petrie, G. Soules, and N. Weiss. 1970. *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*. Annals of Mathematical Statistics. Vol 41, pages 164-171.

M. Becker, W. Drożdżyński, H. Krieger, J. Piskorski, U. Schaefer, and F. Xu. 2002. *SproUT – Shallow Processing with Unification and Typed Feature Structures*. In Proc. of International Conference on NLP (ICON' 2002). Mumbai, India.

F. Bond and C. Vatikiotis-Bateson. 2002. *Using an Ontology to Determine English Countability*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 99-105. Taipei, Taiwan.

K. Bontcheva, D. Maynard, V. Tablan, and H. Cunningham. 2003. *GATE: A Unicode-based Infrastructure Supporting Multilingual Information Extraction*. In Proc. of Workshop on Information Extraction for Slavonic and other Central and Eastern European Languages. Held in conjunction with the 4th International Conference "Recent Advances in Natural Language Processing" (RANLP '2003), Bulgaria.

T. Booth. 1969. *Probabilistic representation of formal language*. In Proc. of 10th Annual IEEE Symposium on Switching and Automata Theory, pages 74-81.

T. Booth and R. Thomson. 1973. *Applying probability measures to abstract languages*. IEEE Transactions on Computers. C-22, pages 442-450.

S. Brennan, M. Friedman, and C. Pollard. 1987. *A centering approach to pronouns*. In Proc. 25th Annual Meeting of the ACL, pages 155-162, Stanford, USA.

D. Brickley and R.V. Guha. 2000. *Resource Description Framework (RDF) Schema Specification 1.0*. World Wide Web Consortium: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.

E. Brill. 1993a. *Automatic grammar induction and parsing free text: A transformation-based approach*. In Proc. of the 31st Meeting of the Association of Computational Linguistics, Columbus, Ohio. USA.

E. Brill. 1993b. *Transformation-based error-driven parsing*. In Proceedings of the Third International Workshop on Parsing Technologies, pages 13-25. Tilburg, The Netherlands.

E. Brill. 1994. *Some advances in transformation-based part of speech tagging.* In Proceedings of the Twelfth National Conference on Artificial Intelligence, pages 722-727. Seattle, Washington.

E. Brill. 1995. *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging.* In Computational Linguistics, Vol. 21, No. 4, pages 543-565.

E. Brill and P. Resnik. 1994. *A rule-based approach to prepositional phrase attachment disambiguation*. In Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING-1994), pages 998-1004. Tyoto, Japan.

M. Califf and R. Mooney. 1999. *Relational Learning of Pattern-Match Rules for Information Extraction*. In Proc. of the 16th National Conference on Artificial Intelligence (AAAI-99), pages 328-334. Orlando, Florida, USA.

Y. Cao. 2001. *The 20th Anniversary of CIPSC*. Proc. of Conference of the 20th Anniversary of Chinese Information Processing Society of China. Press of Tsinghua University, Beijing, China. (In Chinese)

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.

H. Chen. 1993. *Anaphora Resolution Algorithms for Mandarin Chinese*. Communication of Chinese and Oriental Languages Information Processing Society, Vol. 3, No. 2, pages 59-67. Singapore.

H. Chen, Y. Ding, S. Tsai and G. Bian. 1998. *Description of the NTU System Used for MET2*. In Proc. of the Seventh Message Understanding Conference (MUC-7). SAIC. Fairfax, Virginia, USA. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

K. Chen and C. Chen. 2000. *Knowledge Extraction for Identification of Chinese Organization Names*. Proc. of the Second Chinese Processing Workshop. Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong, China.

N. Chinchor. 1997. *MUC-7 Named Entity Task Definition (version 3.5)*. In Proc. of Message Understanding Conference. http://www.icl.pku.edu.cn/bswen/nlp/www.muc.saic.com/ne_task.html.

N. Chinchor. 1998. *Overview of MUC-7/MET-2*. In Proc. of the Seventh Message Understanding Conference (MUC-7). SAIC. Fairfax, Virginia, USA. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Chinese Language Reform Commission of China. 1958. *Chinese Phoneticisation Scheme*. http://www.edu.cn/20011114/3009777.shtml. (In Chinese)

K. Church. 1980. *On Memory Limitations in Natural Language Processing*. MIT/LCS/TR-245. Laboratory for Computer Science, Massachusetts Institute of Technology. Massachusetts, USA.

D. Connolly, F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. 2001. *DAML+OIL (March 2001) Reference Description*. World Wide Web Consortium: http://www.w3.org/TR/daml+oil-reference.

C. Cortes and V. Vapnik. 1995. *Support-vector networks*. Machine Learning. Vol. 20, No. 3, pages 273-297.

T. Cover and P. Hart. 1967. *Nearest neighbor pattern classification*. Institute of Electrical and Enginees Transactions on Information Theory. Vol. 13, pages 21-27.

J. Cowan and D. Sharp. 1988a. *Neural nets*. Quarterly Reviews of Biophysics. Vol 21, pages 365-427.

J. Cowan and D. Sharp. 1988b. *Neural nets and artificial intelligence*. Daedalus. Vol. 117, pages 85-121.

J. Cowie, T. Wakao, L. Guthrie, and W. Jin, 1993. *The Diderot Information Extraction System*. In the First Conference of the Pacific Association for Computational Linguistics, pages 5-14, Harbour Center, Campus of SFU, Vancouver, Canada.

M. Craven. 1999. *Learning to Extract Relations from MEDLINE*. In Proc. of AAAI-99 Workshop on
Machine Learning for Information Extraction. Orlando, Florida, USA.

R. Cullingford. 1978. *Script Application: Computer Understanding of Newspaper Stories*. PhD. Thesis, Technical Report 116. Dept. of Computer Science, Yale University, New Haven, Connecticut. USA.

H. Cunningham, K. Humphreys, R. Gaizauskas, und Y. Wilks. 1997. *GATE - a TIPSTER – based General Architecture for Text Engineering*. In Proc. of the TIPSTER Text Program (Phase III) 6 Month Workshop. DARPA. Morgan Kaufmann Publishers Inc., California, USA.

H. Cunningham, Y. Wilks, and R. Gaizauskas. 1996. *GATE – a General Architecture for Text Engineering*. In Proc. of the 16th Conference on Computational Linguistics (COLING-96), Copenhagen, Denmark.

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, and J. Zavrel. 1999. *Forgetting exceptions is harmful in language learning*. Machine Learning. Special issue on Machine Learning and Natural Language.

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.

J. Dake. 2003. *Explorations of the speed-accuracy trade-off in Memory Based Learning algorithm*. Technical Report ILK-03-02, ILK, Tilburg University. Tilburg, The Netherlands. http://ilk.kub.nl/~ilk/papers/ilk0302.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.

G. DeJong. 1979. *FRUMP: Fast Reading and Understanding Program*. PhD. Thesis. Dept. of Computer Science, Yale University. New Haven, Connecticut, USA.

B. Denison. 2003. Chinese Language Overview. http://pub82.ezboard.com/fmartialarts82280frm10. showMessage?topicID=1.topic. (In Chinese)

J. S. DeRose. 1988. *Grammatical Category Disambiguation by Statistical Optimization*. In Computational Linguistics, Vol. 14, No. 1, pages 31-39.

P. Devijver and J. Kittler. 1982. *Pattern recognition: A statistical approach*. Prentice-Hall, London, UK.

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

R. Duda and P. Hart. 1973. *Pattern Classification and Scene Analysis*. New York: Wiley & Sons.

S. Dudani. 1976. *The distance-weighted k-nearest neighbour rule*. In IEEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-6, pages 325-327.

Y. Freund and R. Schapire. 1999. *Large margin classification using the perception algorithm*. Machine Learning. Vol. 37, No. 3, pages 277-296.

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. 1995. *University of Sheffield: Description of the LaSIE System as used for MUC-6*. In Proc. of the Sixth Message Understanding Conference (MUC-6). Morgan Kaufmann Publishers Inc., California, USA.

R. Gaizauskas, K. Humphreys, S. Azzam, and Y. Wilks. 1997. *Conception vs. Lexicons: An Architecture for Multilingual Information Extraction.* In Maria Teresa Pazienza (Ed.): Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, pages 28-43. Springer-Verlag, Berlin, Germany.

K. Gan and P. Wong. 2000. *Annotating Information Structure in Chinese Texts Using HowNet*. Proc. of the Second Chinese Processing Workshop. Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong, China.

J. Gao, M. Li, and C. Huang. 2003. *Improved Source-Channel Models for Chinese Word Segmentation*. In Proc. of the 41th Annual Meeting of Association for Computational Linguistics. (ACL 2003), pages 272-279. Sapporo, Japan.

I. Good. 1961. *A causal calculus*. British Journal of the Philosophy of Science. Vol 11, pages 305-318.

R. Grishman. 1996. *TIPSTER Architecture Design Document Version 2.2*. Technical Report, DARPA. USA. http://www.tipster.org/.

R. Grishman and B. Sundheim. 1996. *Message Understanding Conference - 6: A Brief History*. In Proc. Of the 16th International Conference on Computational Linguistics. Copenhagen, Denmark.

R. Grishman, J. Hobbs, E. Hovy, A. Sanfilippo, and Y. Wilks. 1999. Cross-lingual Information Extraction and Automated Text Summarization. In E. Hovy, N. Ide, and R. Frederking (eds.) : Multilingual Information Management: Current Levels and Future Abilities. A report Commissioned by the US National Science Foundation. http://www.cs.cmu.edu/~ref/mlim/index.html.

B. Grosz, A. Joshi, and S. Weinstein. 1995. *Centering: A Framework for Modelling the Local Coherence of Discourse*. Computational Linguistics, Vol. 21, No. 2, pages 203-226.

P. Halsall. 2003. History of Chiense Language. http://isis.csuhayward.edu/ALSS/MLL/chinese /chinese /Study/history.html. (In Chinese)

F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. 2003. *OWL Web Ontology Language Reference*. World Wide Web Consortium: http://www.w3.org/TR/2003/WD-owl-ref-20030331/.

E. Harold. 1998. *XML: Extensible Markup Language*. IDG Books Worldwide, Inc. California, USA.

J. Hobbs, D. Appelt, J. Bear, M. Tyson, and D. Magerman. 1991. *The TACITUS System: The MUC-3 Experience*. SRI Technical Note 511, SRI International, Menlo Park, California. USA.

J. Hockenmaier and C. Brew. 1998. *Error-Driven Learning of Chinese Word Segmentation.* In Communications of COLIPS 8 (1), pages 69-84. Singapore.

J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. 1996. *FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text*. In E. Roche and Y. Schabes, editors, Finite State Devices for Natural Language Processing. MIT Press, Cambridge, Massachusetts, USA.

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. 1998. *University of Sheffield: Description on the LaSIE-II System as Used for MUC-7*. In Proc. of the Seventh Message Understanding Conference (MUC-7). SAIC. Fairfax, Virginia, USA. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Y. Jia. 1999. Chinese Semantics. The Press of Beijing University. Beijing, China. (In Chinese)

S. Johansson, E. Atwell, R. Garside, and G. Leech. 1986. *The Tagged LOB Corpus*. Norwegian Computing Centre for the Humanities, Bergen, Norway.

G. John, R. Kohavi, and K. Pfleger. 1994. *Irrelevant features and the subset selection problem*. In Machine Learning: Proc. of the Eleventh International Conference. Morgan Kaufmann, pages 121-129.

A. Joshi and B. Srinivas. 1994. *Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing*. In Proc. of the 15th International Conference on Computational Linguistics (COLING'94), pages 154-160. Kyoto University, Japan.

H. Kucera and W. Francis. 1967. *Computational Analysis of Present-Day American English*. Brown University Press. Providence, Rhode Island, USA.

Y. Lai, R. Wang and W. Hsu. 2002. *A DAML+OIL-Compliant Chinese Lexical Ontology*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 1238-1242. Taipei, Taiwan.

G. Leech. 1987. *Semantics*. Shanghai Foreign Language Education Press, Shanghai, China. (Chinese Version)

B. Li. 1995. *Football Dictionary*. Shanghai Dictionary Press, Shanghai, China. (In Chinese)

C. Li and S. Thompson. 1981. *Mandarin Chinese – A Functional Reference Grammar*. University of California Press. Berkeley and Los Angeles, California, USA.

X. Lin, L. Wang, and D. Sun. 1994. *Dictionary of Verbs in Contemporary Chinese*. Beijing Language and Culture University Press. Beijing China. (In Chinese)

K. Liu. 2000. *Automatic Segmentation and Tagging for Chinese Text*. The Commercial Press, Beijing, China. (In Chinese)

K. Liu. 2001. *Research of Automatic Chinese Word Segmentation*. In Proc. of International Workshop ILT&CIP 2001 on Innovative Language Technology and Chinese Information Processing. Shanghai, China.

Q. Liu and S. Li. 2002. *Word Similarity Computing Based on How-net*. International Journal of Computational Linguistics and Chinese Language Processing, Vol.7, No.2, pages 59-76. Association of Computational Linguistics for Chinese Languages, Taipei, Taiwan.

S. Luke, L. Spector, D. Rager, and J. Hendler. 1997. *Ontology-based Web Agents*. In Johnson, W.L. and Hayes-Roth, B., editors, Proc. of the First International Conference on Autonomous Agents (Agents '97), pages 59-68, ACM Press. Marina del Rey, California, USA.

S. Lü. 1985. Pre-Modern Chinese Demonstrative Pronoun. Xue Lin Press, Shanghai, China.

J. Mei, Y. Zhu, Y. Gao, and H. Yin. 1983. *Tongyici Cilin (A Chinese Thesaurus)*. Shanghai Dictionary Press, Shanghai, China. (In Chinese)

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. *Introduction to WordNet: An on-line lexical database*. Journal of Lexicography, Vol. 3, No. 4, pages 235-244.

A. Mitchell et al. 2002. *Annotation Guidelines for Relation Detection and Characterization (RDC) Version 3.6*. Linguistic Data Consortium. Philadelphia, USA. http://www.ldc.upenn.edu/Projects/ACE.

V. Ng and C. Cardie. 2002. *Improving Machine Learning Approaches to Coreference Resolution*. In Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 104-111. Philadelphia, USA.

G. Ngai, M. Carpuat, and P. Fung. 2002. *Identifying Concepts Across Languages: A First Step towards a Corpus-based Approach to Automatic Ontology Alignment*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 737-743. Taipei, Taiwan.

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.

N. Noy and D. McGuinness. 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Stanford University, Stanford, USA.

D. Palmer. 1997. *A Trainable Rule-Based Algorithm for Word Segmentation*. In Proc. of the 35th Annual Meeting of the Association for Computational Linguistics (ACL '97), Madrid, Spain.

F. Pereira and D. Warren. 1980. *Definite Clause Grammars for Language Analysis – A Survey of the Formalism and a Comparison with Augmented Transition Networks*. Artificial Intelligence, Vol. 13, No. 3, pages 231-278.

J. Piskorski. 2002. *The DFKI finite-state machine toolkit*. Research Report RR-02-04, German Research Center for Artificial Intelligence (DFKI), Saarbruecken, Germany.

J. Piskorski, W. Drożdżyński, F. Xu, and O. Scherf. 2002. *A Flexible XML-based Regular Compiler for Creation and Conversion of Linguistic Resources*. In Proc. of the Third International Conference on Language Resources and Evaluation (LREC-2002). Las Palmas, Gran Canaria, Spain.

N. Qian et al. 1995. *Chinese Linguistics*. The Press of Beijing Language Institute. Beijing, China. (In Chinese)

J. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, USA.

D. Roth and W. Yih. 2002. *Probabilistic Reasoning for Entity & Relation Recognition*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 835-840. Taipei, Taiwan.

N. Sager. 1970. *The Sublanguage Method in String Grammars*. In R. Ewton, Jr. and J. Ornstein (eds.), Studies in Language and Linguistics (89-98).

G. Salton, A. Wong, and C. Yang. 1975. *A vector space model for automatic indexing*. Communications of the ACM, Vol 18, No. 11, pages 613-620.

R. Schank and R. Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates. Potomac. Maryland, USA.

R. Schapire and Y. Singer. 2000. *BoosTexter: A System for Multiclass Multi-label Text Categorization*. Machine Learning. Vol 39, No. 2/3, pages 135-168.

C. Shannon. 1948. *A mathematical theory of communication*. The Bell System Technical Journal. Vol. 27, pages 379-423, 623-656.

L. Shen and J. Chen. 2002. *Using Supertag in MUC-7 Template Relation Task*. Technical Report, MS-CIS-02-26, CIS Dept., University of Pennsylvania, Philadelphia, USA. http://www.cis.upenn.edu/~libin/paper/

S. Soderland. 1996. *CRYSTAL: Learning Domain-specific Text Analysis Rules*. Technical Report. Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, Massachusetts, USA. http://www-nlp.cs.umass.edu/ciir-pubs/te-43.pdf.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. *CRYSTAL: Inducing a Conceptual Dictionary*. In Proc. of the 14th International Joint Conference on Artificial Inteligence (IJCAI '95), pages 1314-1319. Quebec, Canada.

R. Sproat and T. Emerson. 2003. *The First International Chinese Word Segmentation Bakeoff*. In Proc. of the Second SIGHAN Workshop on Chinese Language Processing (ACL 2003 Workshop), 133-143. Sapporo, Japan.

B. Srinivas. 1997. *Complexity of Lexical Descriptions and its Relevance to Partial Parsing*. PhD thesis. University of Pennsylvania, Philadelphia, USA. http://www.cis.upenn.edu/~mickeyc/stag/supertags.html.

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

Stanford Medical Informatics. 2003. *The Protégé Ontology Editor and Knowledge Acquisition System*. the School of Medicine, Stanford University. Stanford, USA. http://protege.stanford.edu/.

Sun Microsystems, Inc. 2003. *Java 2 Platform, Standard Edition (J2SE)*. http://java.sun.com/j2se/1.4.1/index.html.

J. Sun, J. Gao, L. Zhang, M. Zhou, and C. Huang. 2002. *Chinese Named Entity Identification Using Class-based Language Model*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 967-973. Taipei, Taiwan.

E. Tello. 1989. *Object-Oriented Programming for Artificial Intelligence: A Guide to Tools and System Design*. 1st edition. Addison-Wesley Longman Publishing Co., Inc., Boston, USA.

The Editorial Office of Zhonghua Book Company. 1915. A Great Dictionary of Chinese Characters. Zhonghua Book Company, Shanghai, China. (In Chinese)

The Language Institute of Chinese Academy of Social Sciences. 1999. Modern Chinese Dictionary (Revision). The Commercial Press, Beijing, China. (In Chinese)

The Ministry of Mechanics and Electronics Industry of China. 1992. *Contemporary Chinese Language Word Segmentation Specification for Information Processing*. National Standard, The Ministry of Mechanics and Electronics Industry of China. (In Chinese)

M. Uschold and M. Gruninger. 1996. *Ontologies: Principles, Methods and Applications*. AIAI-TR-191, Artificial Intelligence Applications Institute, The University of Edinburgh, Edinburgh, UK.

A. Viterbi. 1967. *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. IEEE Transactions on Information Theory IT-13: pages 1260-1269.

N. Wang and X. Zhou et al. 1999. *Vocabulary Applied General Rules*. The Chun Feng Literature and Art Press, Shen Yang, China. (In Chinese)

D. Wettschereck. 1994. *A Study of Distance-based Machine Learning Algorithms*. PhD thesis, Computer Science Department, Oregon State University, Corvallis, USA.

A. White and W. Liu. 1994. *Bias in information-based measures in decision tree induction*. Machine Learning, Vol. 15, No. 3, pages 321-329.

R. Wilensky. 1978. *Understanding Goal-Based Stories*. PhD. Thesis. Dept. of Computer Science, Yale University. New Haven, Connecticut, USA.

A. Winbald, S. D. Edwards, D. R. King. 1990. *Object-Oriented Software.* Addison Wesley Publishing Company, Inc., Boston, USA.

K. Wong, W. Li and C. Yuan. 1999. *Classifying Temporal Concepts in Chinese for Information Extraction*. In Proc. of the Natural Language Processing Pacific Rim Symposium 1999 (NLPRS '99). Beijing, China.

P. Wong and P. Fung. 2002. *Nouns in WordNet and HowNet: An Analysis and Comparison of Semantic Relations*. In Proc. of the 1st International Conference on Global Wordnet, Mysore, India.

World Wide Web Consortium. 2001. *XML Schema*. World Wide Web Consortium. http://www.w3.org/TR/xmlschema-1/ and http://www.w3.org/TR/xmlschema-2/. MIT, ERCIM, KEIO.

S. Wright. 1921. *Correlation and causation*. Journal of Agricultural Research. Vol 20, pages 557-585.

S. Wright. 1934. *The method of path coefficients*. Annuals of Mathematical Statistics. Vol 5, pages 161-215.

S. Wu and W. Hsu. 2002. *SOAT: A Semi-Automatic Domain Ontology Acquisition Tool from Chinese Corpus*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 1313-1317. Taipei, Taiwan.

W. Wu. 1999. *Chinese Computational Semantics: Relations, Relation Semantic Field and Formal Analysis*. Electronic Industry Press, Beijing, China. (In Chinese)

T. Yao, W. Ding, and G. Erbach. 2002a. *Correcting Word Segmentation and Part-Of-Speech Tagging Errors for Chinese Named Entity Recognition*. In Günter Hommel and Sheng Huanye (Eds.): The Internet Challenge: Technology and Applications, pages 29-36. Kluwer Academic Publishers. Dordrecht, The Netherlands.

T. Yao, W. Ding, and G. Erbach. 2002b. *Repairing Errors for Chinese Word Segmentation and Part-of-Speech Tagging.* In Proc. of the First International Conference on Machine Learning and Cybernetics 2002 (ICMLC 2002), pages 1881-1886. Beijing, China.

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.

S. Ye, T. Chua, and J. Liu. 2002. *An Agent-based Approach to Chinese Named Entity Recognition*. In Proc. of the 19th International Conference on Computational Linguistics (COLING 2002), pages 1149-1155. Taipei, Taiwan.

J. Yu. 1998. *Knowledge Handbook of World Football*. Shanghai San Lian Bookstore. Shanghai, China. (In Chinese)

S. Yu, S. Bai, and P. Wu. 1998. *Description of the Kent Ridge Digital Labs System Used for MUC-7*. In Proc. of the Seventh Message Understanding Conference (MUC-7). SAIC. Fairfax, Virginia, USA. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Y. Yuan. 2002. *On the Semantic Knowledge Resources for Information Extraction*. Journal of Chinese Information Processing. Vol. 16, No. 5. Beijing, China. (In Chinese)

J. Zavrel. 1997. *An empirical re-examination of weighted voting for k-nn*. In W. Daelemans, P. Flach, and A. van den Bosch, editors, Proc. of the 7th Belgian-Dutch Conference on Machine Learning, pages 139-148. Tilburg, The Netherlands.

D. Zelenko, C. Aone, and A. Richardella. 2002. *Kernel Methods for Relation Extraction*. In Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 71-78. Philadelphia, USA.

H. Zhang, Q. Liu, X. Cheng, H. Zhang, and H. Yu. 2003. *Chinese Lexical Analysis Using Hierarchical Hidden Markov Model*. In Proc. of the Second SIGHAN Workshop on Chinese Language Processing (ACL 2003 Workshop), pages 63-70. Sapporo, Japan.

Y. Zhang. 1998. *Chinese Text Interpretation based on Hybrid Method*. PhD Thesis. Shanghai Jiao Tong University, Shanghai, China. (In Chinese)

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.

Q. Zhou and S. Feng. 2000. *Build a relation network representation for How-net*. Journal of Chinese Information Processing. Vol. 14, No. 6. Beijing, China. (In Chinese)

J. Zhu and T. Yao. 1998. *Chinese Information Automatic Extraction*. Journal of Northeastern University (Natural Science). Vol. 19, No. 1., pages 52-54. (In Chinese)

X. Zhu, M. Li, J. Gao, and C. Huang. 2003. *Single Character Chinese Named Entity Recognition*. In Proc. of the Second SIGHAN Workshop on Chinese Language Processing (ACL 2003 Workshop), pages 125-132. Sapporo, Japan.