國立交通大學

資訊工程學系

碩士論文

多層架構之中文具名實體辨識

A Multi-Layered Framework for Chinese Named Entity Recognition

研 究 生：陳大任

指導教授：李錫堅 教授

中華民國九十三年五月

多層架構之中文具名實體辨識

A Multi-Layered Framework for Chinese Named Entity Recognition

研究生：陳大任　　Student：Conrad Chen

指導教授：李錫堅　　Advisor：Hsi-Jian Lee

國立交通大學
資訊工程系
碩士論文

A Thesis
Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science and Information Engineering

May 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年五月

多層架構之中文具名實體辨識

學生：陳大任　　　　　　　　　　　　　　　　指導教授：李錫堅

國立交通大學資訊工程學系碩士班

摘　　　要

未知詞（out-of-vocabulary, OOV）的處理已成為高品質詞法分析（lexical analysis）不可或缺的關鍵之一。在所有未知詞中，具名實體（named entity, NE）不但是最多產的一種，也幾乎沒有產生規律可言，卻通常又是語句中最具意義的部分（人、事、時、地、物）。在這篇論文中，我們設計了一套對於中文具名實體的分類方法，並提出以「產生、濾除、回復」的多層架構來處理中文具名實體的辨識問題。本系統首先以一組統計模型及估算方法，盡量產生所有可能的候選具名實體，以取得高召回率（recall） 接著將謬誤濾除當作模稜問題（ambiguity resolution）來處理，我們使用以最大匹配法（maximal-matching）為主的詞法分析器來解決模稜問題。最後，我們用文樣比對（pattern matching）來偵測前兩個階段所產成的異常錯誤，並加以回復。

我們的系統僅使用純粹字面資訊，並且在人名上取得 96%的高召回率，在譯名、地名、組織名的召回率上，也分別取得令人滿意的 88%、89%、與 80%。整體來說本系統的準確度（precision）超過 90%，排除率（excluding rate）超過 99%；可以說我們僅使用相對較少的資訊，卻得到較好的成效。我們提出的架構

仍保留許多模型設計上的彈性與改進空間。我們可以使用更精確的語言模型、更周詳的估算規則、加入更多的資訊與模型等；以在此架構下，達到最佳的效能。

A Multi-Layered Framework for Chinese Named Entity Recognition

Student：Conrad Chen Advisors：Dr. Hsi-Jian Lee

Department of Computer Science and Information Engineering
National Chiao Tung University

ABSTRACT

The handling of out-of-vocabulary (OOV) words is one of the key points to high performance lexical analysis in natural language processing. Among all OOV words, named entities (NE) are the most productive ones and nearly no generation rules for them exist. Named entities generally constitute the most meaningful parts of sentences (persons, affairs, time, places, and objects). In this paper, we propose a classification of Chinese NEs and a multi-layered “generation, filtering, and recovery” framework to address the NER problem. In our system, a set of statistical models and heuristic rules are first used to generate all possible NE candidates to obtain a high recall rate. Then we treat the candidates filtering as an ambiguity resolution problem. To resolve the ambiguities, we adopt a maximal-matching-rule-driven lexical analyzer. Last, a rule-driven pattern matching method is applied to detect and recover abnormalities in the results of the previous two phases.

Pure lexical information is exploited in our system. We get a high recall rate of 96% with personal names (PER), satisfiable recall rates of 88%, 89%, and 80% with

transliteration names (TRA), location names (LOC), and organization names (ORG), respectively. The overall precision is over 90% and the excluding rate is over 99%. Our system exploits relatively simple information and obtains good performances. Our framework retains much flexibility for the refinement of the model design. There is still a lot of room for improvements. More precise language models could be adopted; more complete heuristic rules could be applied; and more knowledge and information could be added to achieve the ultimate performance under this framework.

誌謝　Acknowledgement

謝謝指導教授給我指引，
謝謝雙親予我養育，
謝謝最愛的女友陪在身旁，
謝謝好兄弟們適時的嘲笑與鼓勵；

謝謝新竹的風，謝謝台北的雨，
謝謝鬱悶時的球賽與音樂，
謝謝每夜陪我回家的路燈和永遠等待主人的座車，
謝謝這個世界，以及所有用心奉獻的人。

Thank my advisor for his guiding,
Thank my parents for their bringing up,
Thank my dear girlfriend for her accompanying,
Thank my buddies for their ridiculing and working up;

Thank the wind in Hsinchu, thank the rain in Taipei,
Thank the music and sports game in my depressing time,
Thank streetlamps nightly homing with my shadow, and the car always waiting there,
Thank the whole world, and everybody dedicating his life.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Words are generally the basic unit to process natural languages. Identifying words in sentences, i.e. lexical analysis, is the first work in computer natural language processing. In most Indo-European languages, such as English and French, there are spaces delimiting words. Lexical analysis of such languages is hence trivial and straightforward. However, in many oriental languages, such as Chinese and Japanese, sentences are composed of string of characters without any delimiters to mark word boundaries. To process these languages, sentences must be segmented into word sequences first.

Most Chinese language processing systems rely on lexicons to recognize words in sentences. Because the number of Chinese words is tremendous, it is impossible to compile all Chinese words in a lexicon. Therefore, word segmentation processes would encounter the problem of out-of-vocabulary (OOV) words. In recent years, many researchers start to realize that the identification of OOV words is a key to the success of high precision segmentation [7]. In previous researches, it was assumed that all words in the input documents are known, i.e. there are no OOV words in the documents, and almost all proposed lexical analysis methods could achieve a high accuracy under this assumption, such as [2, 6, 9, 35, and 41]. However, when OOV words were present, the performances were degraded a lot.

Named entities are one of the most important sorts of OOV words. It is impossible to list them exhaustively in a lexicon. Since they are the most productive type of words, nearly no simple or unified generation rules for them exist. Besides, they are usually keywords in documents. Thus, named entity recognition (NER) becomes a major task to many natural language applications, such as natural language understanding, question answering, and information retrieval. Figure 1.1 shows some sentences with named entities underlined:

時報文化出版張大春的城邦暴力團。
麥可傑克遜去佛羅里達的迪斯奈樂園玩。
中華民國國父孫中山的誕辰是十一月十二號。
小李昨天去頂好超市買了五個 富士蘋果。
天湘園的左宗棠雞很好吃。
臥龍先生用八陣圖困住吳將陸遜。

Fig. 1.1 Sample sentences with named entities.

This paper proposes a multi-layered “generation, filtering, and recovery” framework to solve the NER problem. In the generation phase, several stochastic models are responsible for generating all possible candidates of different kinds of named entities in input documents. In the filtering phase, a rule-driven lexical analysis is performed to filter out over-generated false candidates produced in the previous phase. In the recovery phase, segmentation results of lexical analysis are examined by a rule-driven lexical checker to recover the failures of the above two phases.

In Chapter 2 we detail our classification and definition of Chinese NEs. Chapter 3 explains the basic concept and the structure of our system. The candidate generation is discussed in Chapter 4 and the other two phases are arranged in Chapter 5. Chapter 6 gives the result of our experiments and Chapter 7 explains our conclusions.

1.1. Background

Words could be roughly classified into four categories: common words, derived words, named entities, and technical terms for specific domains. No matter how a computer lexicon is constructed, relying on corpora or using existing paper-printed dictionaries, the resulting lexicon generally contains most of common words, some often seen derived words and named entities, and few domain terms. However, OOV words always exist. Derived words, named entities, and domain terms usually constitute the three major part of OOV words, as depicted in Figure 1.2. Notice that there are overlapped areas between these categories; for example, the word "上壘" is both a derived word and a domain term in baseball. The discussions about the three major types of OOV words will be detailed separately in the following sections.

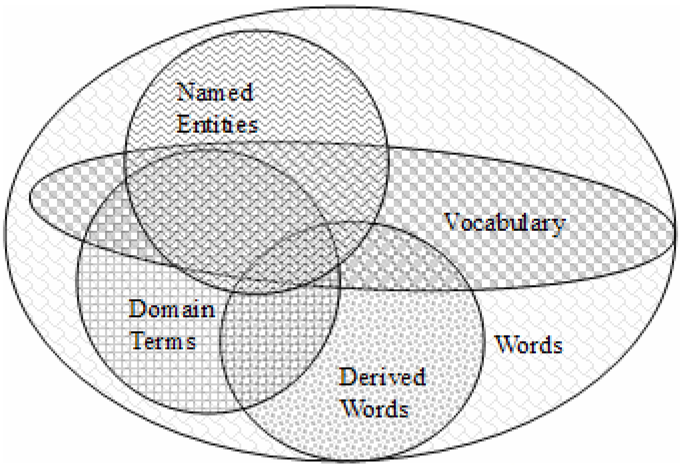


Fig. 1.2 A rough category of Chinese words.

1.1.1. Domain Terms

A domain term is defined as a word or an expression that designates a specific concept in a special domain [20]. For example, "和弦(chord)" is a term in the music domain. Most domain terms are frequently used in their specific domain but hardly seen in other domains. A few domain terms are extended to more general senses, such as "開刀(to operate)" and "出局(out) ."

Domain terms can be recognized by consulting a domain lexicon, which can be extracted from a domain specific corpus in a systematic way [10]. Generally it belongs to the territory of offline new-word extraction and it is not the focus of this paper.

1.1.2. Derived Words

The definition of derived words in Chinese is slightly different from that of English. There are four major ways in which derived words may be produced [36]:

- reduplication (e.g. 高高興興, 看一看…)
- affixation (e.g. 一般性, 烘乾機…),
- directional and resultant compounding (e.g. 進去, 做完…)
- merging and splitting (e.g. 國內外, 吃個飯…)

Recognition of derived words usually relies on handcrafted or corpus-retrieved morphological rules and semantic classification of words [7]. However, there is no standard for how to segment derived words. Some researches regard derived words as a single word (e.g. Huang et al. (1997) [17]), but the others tend to treat them as a combination of normal words and affixes (e.g. GB/T 13715-92 (1993) [15]). For example, the derived word "一般性" may be segmented as "/一般性/" or "/一般/性/" in different systems. It is hard to say which choice is better or more correct. In different application domains, each has its advantages and disadvantages [36]. Even experts or native speakers cannot have an agreement on it [28].

1.1.3. Named Entities

The definition of named entities is not trivial. Roughly speaking, a named entity is either a numerical expression or a proper noun that denotes a person, an organization, a location, a product, etc. In other words, a name entity refers to a unique identity; instead, a common word represents a generic class. For example, "史努比(Snoopy)" is a named entity, but "狗(dog)" is a common word.

Named entities comprise many different types. The Message Understanding

Conference (MUC) has divided named entity task into five subtasks: persons, locations, organizations, times, and quantities [11]. This classification is mainly used as an evaluation criterion; it is not detailed enough to cover all kinds of name entities. Sakine [25] has proposed a hierarchy of Japanese named entity classification, which contains 150 subclasses. In this thesis, we modify Sakine's classification to make it suitable for Chinese. The complete classification hierarchy is given in Appendix A.

There are nine major classes in our classification; seven of them are open-ended:

{persons, locations, organizations, terminologies, titles, miscellanea, abbreviations}

And two of them are close-ended:

{times, quantities}

Open-ended named entities can be further divided into two types:

(original | translated)

Each class will be discussed individually in detail in Chapter 2.

1.2 Related Works

Many researches have addressed the NE recognition problem in Chinese since 1990. However, most of them have covered only a few types of NEs. They generally focused on three main open-ended major classes (PER, LOC, and ORG) defined in MUC information extraction task plus transliteration names (TRA) (which denotes names translated in phonetic values) only.

Generally speaking, there are no well-accept standards for NE recognition, hence most of these researchers followed their own definitions of NEs and the results thus obtained are less comparable.

Since researchers usually take different ways to deal with different types of NE, the following sections are organized according to the focusing NE types of researches.

Section 1.2.1 discussed personal name recognition methods. Section 1.2.2~1.2.4 cover LOC, ORG, and TRA recognition approaches respectively. Several systems employed type-independent methods. These systems will be presented in Section 1.2.5.

1.2.1 Personal Name Recognition

The identification of personal names is one of the most popular and earliest explored topics about NER. [3, 28, and 44] took purely statistical methods to recognize unknown personal names in documents; on the other hand, [5, 18, 27, 31, and 32] took hybrid ways of segmentation, stochastic model, contextual information and heuristic rules. Above models and their performance reported will be detailed in following paragraphs.

- Statistic Approaches

Chang et al. (1992) [3] used purely statistical approach and got a 91.87% precision and 80.67% recall. Sproat *et al.* (1996) [28] based on this method and improve the estimation method of statistics. Zheng et al. (2002) [44] used the regular probabilistic grammar to identify candidates and the statistical segmentation to verify these candidates. They got a recall rate of 75.17% and a precision of 90.98%.

- Hybrid Approaches

Song et al. (1993) [27] used segmentation and rule-driven approach and achieved 94.7% recall and 97.4% precision. Sun et al. (1994) [31] took a similar approach, but added probability and other checking mechanism. They got 99.77% recall but a lower precision 70.06%. Chen et al (1996) [5] combined pre-segmentation, statistical models, and several lexical clues, and obtained 88.04% recall and 92.56% precision. Sun et al. (1997) [32] integrated probability model, reduplication patterns, and contextual information. They achieved 95.0% recall and 87.6% precision. Ji & Luo (1999) [18] proposed inverse name frequency (INF) model combined with several heuristic rules and obtained 93.75% recall and 83.95% precision.

1.2.2 Location Name Recognition

The recognition of location names is comparatively less mentioned by previous researchers. Many systems collected frequently used location names, like “台北” and “高雄,” as vocabularies. However, the performance of LOC recognition is generally lower than PER recognition. Usually probability models and contextual information are employed to recognize location names.

Shen et al. (1995) [26] employed pre-segmentation and frequency model to recognize location names and got 95.0% recall and 81.1% precision. Tan et al. (1999) [33] used probabilistic models to extract name candidates and several contextual rules are applied to verify them. They got 93.8% recall and 86.7% precision.

1.2.3 Organization Name Recognition

The structure of organization names is more complex than above two types of NE. Affixes are generally main features used to recognize ORG. Chen & Lee (1994) [4] used morphological rules, mutual information of words, and contextual information to identify the names of organizations. They achieved a recall rate of 54.50% and a precision rate of 61.79%. Chen & Chen (2000) [8] proposed knowledge extraction and pattern matching approach to identify organization names with suffix, and got 92% precision. Wu et al. (2003) [38] employed co-training method to obtain a statistical model, and combined it with several heuristic rules. They got 75.5% recall and 88.5% precision.

1.2.4 Transliteration Name Recognition

Sun (1993) [30] used simple handcrafted rules and transliterating character set to identify the boundaries of transliteration names. They achieved 98% recall and 63% precision. Xiao et al. (2002) [39] proposed a bootstrapping algorithm to extract transliterating characters and contextual information from the Internet. Stochastic models were then used to recognize transliteration names. They achieve 95.6% in recall and 89% in precision.

1.2.5 Type-Independent Approaches

There are many type-independent approaches of NER. However, most of these approaches need type-dependent data, such as frequencies, role tags, and contextual information. Roughly speaking, type independent models can be divided into two major sorts: over-generating & disambiguating [13, 21, 40, and 42] and over-segmenting & generating [16, 29, 37, and 43]. These systems and its performance reported will be detailed in following paragraphs.

- Over-generating & Disambiguating

Yu et al. (1998) [42] employed contextual information and internal features to extract NE candidates and used probabilistic models of part-of-speech tags to perform disambiguation. A similar system is also reported in Luo & Song (2001) [21]. Ye et al. (2002) [40] treated NER as a multi-agent negotiation problem based on probabilistic model, and obtained 96% and 93% (R&P) with PER, 94% and 91% with LOC, and 95% and 93% with ORG. Chua and Liu (2002) [13] proposed a corpus learning and generalizing method of pattern rules. These rules were combined with a hybrid pre-segmentation with NE recognition to generate all possible NEs, and decision trees were used to resolve ambiguities. They obtained 98% and 91% (R&P) with PER, 92% and 90% with LOC, and 92% and 91% with ORG.

■ Over-segmenting & Generating

Most over-segmenting and generating methods are based on class-based role tagging approaches. A segmenting and tagging process is taken first, and then various NE generating methods are applied. Zhang et al. (2002) [43] applied template matching to generate NE candidates and got 69.88% precision and 91.65% recall with personal names, and 77.52% precision and 93.97% recall with transliteration names. Goh et al. (2003) [16] adopted support vector machine to chunk NE candidates and obtained 84.44% recall and 89.25% precision with personal names, and 63.25% recall and 79.36% precision with organization names. Sun et al. (2002) [29] combined their system with heuristic information, cache model and abbreviation model. They achieved 79.86% precision and 87.29% recall with PER, 80.88% precision and 82.46% recall with LOC, and 76.63% precision and 56.54% recall with ORG. Wu et al. (2003) [37] incorporated various human knowledge resources like thesaurus. They obtained 83.30% precision and 92.28% recall with PER, 88.31% precision and 84.69% recall with LOC, and 84.49% precision and 71.08% recall with ORG.

Chapter 2 Classification of Named Entities

In this chapter, we try to propose a classification of Chinese named entities in a natural language processing aspect. In our classification, Chinese NEs are divided into nine subclasses as Figure 2.1 shows:

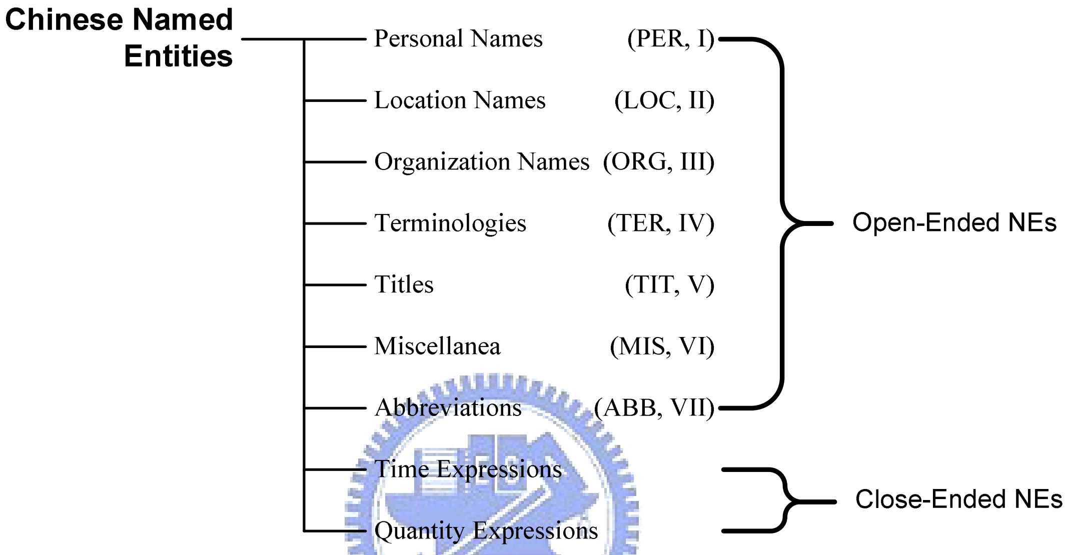


Fig. 2.1 The nine subclasses of Chinese named entities.

We attempt to cover all possible named entities in our classification, and give a preliminary analysis of them. Denotations like (FPN_CHI, I.1.a) appearing in the following sections denote the position of that subclass in the classification hierarchy. Appendix A gives the full classification hierarchy.

In fact, distinctions among some subclasses are vague, such as some sub-types of location names and organization names. Even more, one named entity maybe figures as different types in one sentence. Take the following sentence as an example:

新新聞社長請大家買新新聞回家看

The first appearance of "新新聞" represent an organization, however the second one stands for a publication. Nevertheless, a throughout study of the range and the classification of NE is necessary.

In general, the structure of open-ended NEs except for abbreviations is a genuine name followed by optional suffixes. It can be represented as:

Genuine name + [suffixes]

Suffixes are generally common nouns, which indicate the generic class to which the named entity belongs. For example, “台北” is a *genuine name* and “台北市” is a *whole named entity* with *suffix* “市” indicating that “台北” is a city. There are several disputations about if the *suffix* part should be considered as a part of named entity. As we mentioned before, no unified standard exists at present; even in same corpus, there are still some confusions. For instance, Sinica Corpus [12] viewed “台北市” as a whole named entity, but split “新學友 / 書局” on the contrary.

To follow Bloomfield’s definition of words, “word is the smallest units of speech that can meaningfully stand by their own [1],” we consider a named entity as a maximal qualifier-less string that represents a unique individual. For example, “英代爾公司台灣分公司” is treated as “英代爾公司 / 台灣 / 分公司,” not “英代爾公司台灣分公司” or “英代爾 / 公司 / 台灣 / 分公司.” “公司” is viewed as a *suffix*, and “英代爾公司” is a unique individual. On the other hand, “台灣分公司” is viewed as a qualifier, which indicates that it is a branch of Intel in Taiwan.

As we mentioned in the previous chapter, *genuine names* of open-ended NEs can be further divided into *original* ones and *translated* ones. Strictly speaking, translated names are a feature, not a category standing alone; however the difference among translated names of different types of NEs is little, and the features of them are much varied from original ones. Therefore, an individual section is used to address translated names.

Section 2.1-2.7 states the seven subclasses of original Chinese open-ended NEs, and Section 2.8 addresses translated ones. Close-ended NEs are explained in Section 2.9

2.1. Persons

Personal names (PER, I) are the most important and the most often seen named entities. As implied, it denotes the name of a person, either a formal or informal one. Section 2.1.1 describes formal personal names, and Section 2.1.2 discusses informal ones. Personal names with appellations are stated in Section 2.1.3.

2.1.1. Formal Personal Names

Formal personal names (FORMAL, I.1) mean the names registered officially which represents our identities. Most oriental people have formal names constituted of Chinese characters, or more precisely, "hanzi." The structure of a hanzi name is usually:

Surname + Given Name

(Except Vietnamese, most Vietnamese also have a middle name.) People who use hanzi names include:

- *Chinese* (FOR_CHI, I.1.a)
- *Japanese* (FOR_JAP, I.1.b)
- *Korean* (FOR_KOR, I.1.c)
- *Vietnamese* (FOR_VIE, I.1.d)

When these hanzi names are translated into Chinese, the original characters will be kept, or transcribed into the equivalent characters in Chinese. For example, the Japanese name "徳川家康" will be translated into "德川家康."

The set of possible surnames and given names are quite different from countries to countries. There are at least 100,000 surnames in Japanese, but the number of characters that could constitute a given name is less than 2,000. The number of currently used Chinese surnames is more than 3,000. Except a few words with radicals of bad meanings like “尸,” almost every Chinese character might appear in a given name. On the other hand, in contrast with Japanese and Chinese, there are less than 500 Korean surnames and only about 200 Vietnamese surnames [23].

The length of surnames and given names are varied. The length of Japanese surnames could be varied from one character to nine characters (although most of them are shorter than four characters). Except a few transliterated surnames of minority nationalities (like Mongolian or Formosan Aboriginal), most Chinese surnames are either one or two characters long. All Korean and Vietnamese surnames are shorter than three characters.

The statistic distributions of surnames and given names of these languages are dissimilar. The top ten of most frequently occurring Japanese surnames account for about 20% of Japanese population. In Chinese, the ratio is raised to over 40%. In Korea, the three major surnames (金(Kim), 李(Lee), and 朴(Park)) make up about half of the population, and in Vietnam, for every two people there is one that his surname is “阮(Nguyen).”

Besides the diversity among these languages, even in Chinese, there are different naming customs in different regions. In Mainland China, over 30% of people have a given name of single character, but in Taiwan or Hong Kong, single-character given names are much less. Dialects also influence the choice of given names, too. For example, “俊宏” is a typical Taiwanese given name and “家輝” is a typical Cantonese one.

The differences mentioned above mean that there should be different rules or different statistic features to recognize personal names in different languages, or even in different regions. Therefore, in our classification, formal Chinese names from different regions are split up into their own subclasses.

2.1.2. Informal Personal Names

There are many informal names, or aliases (ALIAS, I.2), used in our daily life. We usually use nicknames (ALI_NIC, I.2.a) to call our family and sobriquets (ALI_SOB, I.2.b) to call our friends. Besides these, some Chinese, especially ancient scholars, may be referred by their pseudonyms (ALI_PSE, I.2.c) instead of their original names, like “青蓮居士” for “李白(Li Bai)” and “適之” for “胡適.”

Not only humans have names, but humanized-characters (CHARAC, I.3), also do. “米老鼠(Mickey Mouse)” is undoubtedly a named entity. Besides cartoon or animals, spirits and monsters (SPIRIT, I.4) have names too, such as “宙斯(Zeus)” and “梅杜莎(Medusa).” There are nearly no restrictions in the morphology of aliases.

2.1.3. Personal Name with Appellation

It is often that appellations are attached after personal names to emphasize roles of somebody, such as “王小姐,” “富雄兄,” and “李家同校長” (APPELL, I.5). Often seen appellations include “先生,” “小姐,” and various nouns representing relationships and occupations.

Should personal names with appellation, like “陳水扁總統(President Chen)” or “李教授(Prof. Li),” be viewed as a single named entity? Both ways have their advantages and disadvantages [7]. In our classification, we follow the principle of maximal string and view them as whole named entities.

2.2. Locations

In our classification, there are four subclasses of location names (LOC, II):

- place names (PLA, II.1)
- geography (GEO, II.2)
- architectures (ARC, II.3)
- addresses (ADD, II.4)

Place names contain three sub-types:

- common place names (PLA_COM, II.1.a)
- complex place names (PLA_CLP, II.1.b)
- administrative divisions (PLA_ADM, II.1.c)

Figure 2.2 shows some sentences with location names underlined:

清水斷崖是指蘇花公路從和仁到崇德間的路段。
白米甕砲台位於基隆市 中正區 港太社區的山頂上，又名光華古砲台。
從新光大樓遠眺臺北與陽明山，夜裡中山高速公路沿基隆河而行。
威尼斯飯店把義大利 威尼斯的聖馬可廣場搬到拉斯維加斯來。
交通大學的地址是新竹市大學路**1001**號。

Fig. 2.2 Sample sentences with location names.

The distinction between common place names and complex place names is their modifying affixes. A complex place name can be represented as:

prefix + place name | place name + suffix

For example, “台灣(Taiwan)” is a common place name, and “南台灣(Southern Taiwan)” is a complex place name. However, we will treat them uniformly in our recognition models, because the boundary between common place names and complex place names are vague in Chinese. For example, “臺北(Taipei)” could be treated as a common place name or a complex place name. If we take all this kind of names as a common place name, and take all these affixes as a common naming character of place name, the problem we face will be much simplified. The often seen affixes include 東, 西, 南, 北, 中, 內, 外, etc.

Most administrative division names are limited in a fixed set. There are many systematic ways to collect the most commonly seen administrative division names. The discriminative features of them from above two types are division suffixes indicating their division levels, such as “縣,” “市,” “鄉,” “鎮,” etc. Besides division names in real world today, there are historic and fabricated ones, of course. For example, “開封府” and “天蔭城” are also division names.

Geographic names include:

- water areas (GEO_WAT, II.2.a)
- terrains (GEO_TER, II.2.b)

For example, “濁水溪” is a water area name and “玉山” is a terrain name. Geographic names are usually composed of a common place name and a suffix representing the geography, such as “基隆河” and “臺北盆地”:

Geographic name: common place name + suffix

The morphological structure of architecture names is similar to geographic names. However, the *genuine name* of architecture names is more diversified. There are three sub-types of architecture names in our classification:

- infrastructures (ARC_INF, II.3.a)
- buildings (ARC_BUI, II.3.b)
- facilities (ARC_FAC, II.3.c)

The distinction among these three types is the most important meaning they represent. Infrastructure names chiefly represent the functionalities they carry, like “中橫公路,” “高雄港,” “九廣鐵路,” etc. Building names chiefly represent the entities themselves, like “新光摩天大樓,” “大雁塔,” “明孝陵,” etc. Facility names chiefly represent the people interacting with them, like “交通大學,” “台大醫院,” “三重分店,” etc.

Not only common place names may appear as *genuine names* of architectures, but also personal names like “中山高速公路,” organization names like “國泰醫院,” and other entity names like “自由女神像” might, too. It is necessary to combining several models to recognize architecture names. Fortunately, the specific suffixes are helpful features.

Addresses are a special type of location names. They are composed of division names, infrastructure names, and quantities:

Address: [division name]$^{+}$ + *[infrastructure name]** + *[quantity expression]*$^{+}$

For example, in the address “新竹市大學路1001號,” “新竹市” is a division name, “大學路” is an infrastructure name, and “1001號” is a quantity expression.

2.3. Organizations

In our classification, there are four major types of organization names (ORG, III):

- governments (GOV, III.1)
- people groups (GRO, III.2)
- races (RAC, III.3)
- companies (COM, III.4)

Figure 2.3 shows some sentences with organization names underlined:

消費者基金會對公平會處理政府與微軟的和解案表示不滿。

庫德族的民族黨與愛國聯盟兩大黨。

中華職棒聯盟冠軍戰由兄弟象迎戰興農牛。

台鐵與鐵路工會各執一詞。

少林、武當、峨眉是三大武術門派。

國軍海鷗部隊和警政署 空中警察隊。

Fig. 2.3 Sample sentences with organization names.

The reason for dividing organization names into four types is the discrepancy in their usages.

The distinction between organizations and locations is that the relationship of people is more important to organizations; however the real place is more important to locations. In other words, the definition of organization in our classification is more “abstract” than location. For example, “誠品書店” is a location, but “誠品股份有限公司” is an organization.

Government names are restricted in a very limited set. The name of governments is more like a common word describing the duty of the department than a name given by people, as “行政院,” “外交部,” “環保局,” etc. They still should be named entities, though. Government names may not be productive in real world, but in novel worlds, it is not just that simple, for instance, the department named “魔法部” in Harry Potter.

People group names cover a vast area. The major subclasses of people groups include:

- associations (GRO_ASS, III.2.a) (e.g. “董氏基金會,” “澎湖觀光協會”)
- parties (GRO_PAR, III.2.b) (e.g. “工黨,” “民進黨”)
- religions (GRO_REG, III.2.c) (e.g. “基督教,” “一貫道”)
- teams (GRO_TEA, III.2.d) (e.g. “兄弟隊,” “明尼蘇達灰狼”)
- gangs (GRO_GAN, III.2.e) (e.g. “天道盟,” “武當派”)
- forces (GRO_FOR III.2.f) (e.g. “黑貓中隊,” “飛虎隊”)

Usually there are suffixes representing the type of them in people group names. However, each kind of people group has its own characteristics and there are too many kinds of people groups to exhaustively analyze their statistic features and internal structures here.

The number of most frequently used racial names is limited. However it’s very easy for novelist to fabricate new racial names. Major suffixes of racial names are “人” and “族.” These two words are also frequently used to make a new derived word indicating some group of people with common characteristics, such as “媒體人” and “飆車族.” That causes a little problem to distinguish racial names from these derived words. .

Structures of company names are different from that of other organization names.

Most organization names generally have appended suffixes, such as “院,” “會,” “族,” etc, but company names do not. Even in more formal documents like newspapers, company names are not often attached with suffixes like “公司” or “工業.” Fortunately, characters that may appear in a company name are not many. In our estimation, about 90% of companies have a name constituted within a 2,000-character set. Nevertheless, unlike personal names, the most frequently used company-naming characters are also the ones that constitute the most frequently used words. It might cause difficulties in company name recognition.

2.4. Terminologies

As implied, terminologies (TER, IV) include the terms in the overlapped area of named entities and domain terms. In academic studies, we often give something (such as laws, theories, and so on) a name for an easier communication. For example, “牛頓第一運動定律(Newton's First Law of Motion)” is more succinct and precise than “在沒有外力作用下，物體靜者恆靜，動者恆動(Every object in a state of uniform motion tends to remain in that state of motion unless an external force is applied to it.).”

Terminologies are diversified and cover many domains. Here we consider the frequently used ones only:

- taxonomic classifications (TAX, IV.1)
- chemical (CHE, IV.2)
- medical (MED, IV.3)
- geosciences (GES, IV.4)
- astronomy (AST, IV.5)
- phenomenon (PHE, IV.6)

- measurements (MEA, IV.7)
- type specifications (TSP, IV.8)

Figure 2.4 shows some sentences with terminologies underlined:

多吃**富士蘋果**攝取**維生素 C** 不會得**敗血病**。
硫酸鋁鉀就是俗稱的**明礬**。
我坐在一顆**大理石**上看著**獵戶座**。
娜莉颱風的動向無法用**牛頓定律**來解釋。
法拉第和**庫倫**都是電學單位。

Fig. 2.4 Sample sentences with terminologies.

Taxonomic classification names are the most often seen terminologies. The boundary between taxonomic names and common words is sometimes vague. For instance, “富士蘋果(Fuji apple)” is undoubtedly a taxonomic name, but the case with “蘋果(apple)” is not so clear. There are also some taxonomic names occurred in novels or fantasies, like “龍(dragon)” and “獨角獸(unicorn).” It is very easy for a novelist to invent a new term. It generally needs a deeper semantic analysis to handle these fabricated names.

In our classification, chemical names include not only formal academic terminologies, but also the common names of chemical instance how we call them in our daily lives. For example, both “硫酸鋁鉀(aluminum potassium sulfate)” and “明礬(alum)” are chemical names. The former could be recognized by pattern matching with handcrafted regular expressions. The latter should be listed in lexicons.

Medical names include the name of drugs, diseases, treatments, medical appliances, etc. Geosciences names include the names of geology, oceanography, atmospheric science, etc. Astronomy names include the names of stars, constellations, galaxies, etc. Phenomenon names include the name of laws, theorems, theories, effects, hypotheses, etc. The names of measurement and type specification (TSP, IV.8) are also classified into terminology. “公分(centimeter)” or “公克(gram)” does not look like a named entity, but “安培(ampere)” and “法拉第(farad)” could give a clearer explanation.

Like other domain terms, generally there are several systematic ways to collect these terminological names, if we put the fabricated names aside.

2.5. Titles

Titles (TIT, V) contain following subclasses:

- publications (PUB, V.1)
- creations (CRE, V.2)
- skills (SKI, V.3)
- styles (STY, V.4)
- cultures (CUL, V.5)
- brands (BRA, V.6)

Figure 2.5 shows some sentences with titles underlined:

一九八四與美麗新世界都是反烏托邦小說。
星空是著名的印象派畫作。
雪碧跟可口可樂由同公司出品。
大英百科全書有介紹馬雅文明。
他跳爵士舞像在打太極拳。
降龍十八掌第一式亢龍有悔。

Fig. 2.5 Sample sentences with titles.

Generally a title is denominated or designated by someone with creativity, so there are nearly no regularity or formation rules for titles. It may be a common word, as “渴望(Aspire®);” it may be a clause, as “挪威的森林(Norwegian Wood);” it may be a sentence, as “阿根廷別爲我哭泣(Don’t Cry for Me Argentina);” it may even seems like nothing, as “可口可樂(Coca-Cola®).”

Most titles of creations and brands don’t have suffixes. This makes it more difficult to recognize them. Unfortunately, creations and brands are also the most important types of titles. There might be two ways to solve the problem caused by non-suffixed titles: deep semantic analysis or built-in common sense database. To the former, as mentioned before, semantic analyzing is a difficult topic itself. To the latter, titles are so productive that a built-in database could only resolve a little part of problem.

2.6. Miscellanea

There are many named entities that cannot be classified into any of the above subclasses. These named entities are not important or frequently used enough to form a new subclass. Thus, we put all of them under the subclass “miscellanea” (MIS, VI).

These miscellaneous named entities include:

- vehicles (ENT_VEH, VI.1.a) (e.g. 空軍一號，五月花號)
- weapons (ENT_WEA, VI.1.b) (e.g. 倚天劍，達姆彈)
- foods (ENT_FOO, VI.1.c) (e.g. 提拉米蘇，卡布其諾)
- cloths and clothes (ENT_CLO, VI.1.d) (e.g. 尼龍，卡其褲)
- festivals (FES, VI.2) (e.g. 復活節，林肯紀念日)
- reign titles (REG, VI.3) (e.g. 都鐸王朝，光緒)
- contests and awards (CON, VI.4) (e.g. 溫布頓公開賽，諾貝爾獎)
- historic events (EVE, VI.5) (e.g. 水門案，甲午戰爭)
- plans (PLN, VI.6) (e.g. 沙漠風暴，春安專案)

Figure 2.6 shows some sentences with miscellaneous names underlined:

小鷹號配備有海麻雀飛彈。
端午節我們家會包福州粽吃。
清朝 光緒年間發生甲午戰爭。
他穿蘇格蘭裙出席奧斯卡獎。

Fig. 2.6 Sample sentences with miscellaneous names.

2.7. Abbreviations

There are three types of abbreviated NEs (ABB, VII) in our classification:

- single entity abbreviations (ABB_SIG, VII.1)
- combined abbreviations (ABB_COM, VII.2)
- derived named entities (ABB_DER, VII.3)

The first type is single entity abbreviations. In other words, it means the abbreviation which denotes only one named entity. The characters composing this type of abbreviations are usually picked from the characters constituting the original named entity. For example, “皇家馬德里” is abbreviated to “皇馬” and “僑務委員會” is abbreviated to “僑委會.” Which characters should be selected generally depends on the inner structure of the original named entity. Always taking the first character of each inner component may obtain an acceptable precision. However, there are many exceptions to this rule; for instance, “行政院” is abbreviated to “政院,” not “行院.” Sometimes a single abbreviation comprises not only the characters of the original entity but also an affix denoting the abbreviating. For example, “陳某” represents somebody with the surname “陳.”

The second type is combined abbreviations. It means the abbreviation denoting more than one named entity. Generally combined abbreviations are composed of one character from each component entity, like “孔孟” denotes “孔子” and “孟子,” “高屏” denotes “高雄” and “屏東.” In other cases, while the most significant or the most meaningful character of each component is same, the combined abbreviation will comprise this common character and an affixed number. For instance, “湖南” and “湖北” are combined to “兩湖,” “蘇洵,” “蘇軾,” and “蘇轍” are combined to “三蘇.”

The third type is derived NEs. It is an overlapped area between derived words and NEs. Sometimes NEs might appear as morphemic components of compounding derived words, such as “林家” and “日本貨.” They might appear as a subject (e.g. “美製”), a modifier (e.g. “美援”), or an object (e.g. “留美”). Besides compounding ones, there are also merged NEs, such as “高雄市政” and a more complicate case “復興忠孝路口.”

There is some confusion with derived NEs. For example, “返台” and “台北市長” could be interpreted as “台北 / 市長” and “返 / 台” respectively, but “石林彝族自治縣長” is difficultly to viewed as “石林彝族自治 / 縣長” or “石林彝族 / 自治縣長.” Splitting it or not is an implementing choice, both two ways have their advantages and disadvantages.

In general, before the first appearance of the abbreviation, the original named entities would be mentioned in advance in documents. Therefore we could generate the possible abbreviations according to the original entities to help the recognition of them. However, often cases are not so simple. Occasionally for some reason (e.g. the document is too long), the full document cannot be processed at once. The original entities maybe appeared long before the abbreviation did, so the relationship between them cannot be found. In other cases, some abbreviations are so popularly used. Writers may use them in documents directly without mentioning the original names first (e.g. 馬立強, 中研院). Without common sense, it is very hard to resolve this kind of abbreviations.

2.8. Translated Names

Translated names cover all subclasses above. There are translated personal names, translated location names, translated organization names, and so forth. With or without suffixes, almost all kinds of named entities may have a translated *genuine name*. Almost all named entities appearing in translated documents are translated ones. Processing named entities without handling translated names is just half way to destination.

There are three ways to translate a foreign name into Chinese. The first, the commonest one, is choosing characters with similar phonetic spelling to the original foreign name, i.e. transliteration. For example, "Peter" becomes "彼得(Bi-De)" and "New York" becomes "紐約(Niu-Yue)."

The second way, mainly occurring with personal names, is to translate the foreign name into a Chinese-style name with more or less similar pronunciation. This type of translation generally occurs in two situations. Firstly, in novel or movie, because it is hard for Chinese to remember a long senseless translated name, in order to relief audience's burden and give a stronger impression on characters in the story, Chinese-style translated names are sometimes introduced. For example, "Scarlet O'Hara," a character in Gone with the Wind, is translated into "郝思嘉(Hao Si-Jia)" instead of "史卡麗特歐哈拉(Shi-Ka-Li-Te O-Ha-La)." On the other hand, some foreigners residing in China like to give themselves a Chinese-style name. For example, the former AIT chairman, Richard Bush, has a Chinese-style name "卜睿哲(Bu Rui-Zhe)."

The third way of translation, translating in meanings, mainly happens with aliases or names of entities other than people. For instance, "Cinderella" is translated to "灰姑娘(Cinder Girl)," "San Francisco" is translated to "舊金山 (Old Golden

Mountain),”and “Microsoft®” is translated to “微軟(Micro Soft).”

2.9. Time & Quantity

Time and quantity expressions could be viewed as another type of derived words other than those four we discussed before. A time or quantity expression usually can be divided into four parts: ordinal prefix, numeral, qualifier, and unit suffix. Take “第三十多名” as example, “第” is the ordinal prefix, “三十” is the numeral part, “多” is the qualifier, and “名” is the unit suffix. There are many types of time and quantity expressions. The main distinctions among them are possible range of numeral values, repeating times of expressions, and order of unit suffixes. For example, “六月七日” is a legal time expression, but “十三月七日,” “六月六月七日,” or “七日六月” are not.

Chapter 3　System Structure

3.1. Basic Concepts

In this paper, we try to imitate human behavior in the recognition of open-ended named entities. (Unless there are specific denotations, “named entity” denotes the open-ended ones in the rest of this paper.) We believe that, following humans’ thought if possible, there are less chances of missing special cases. How do humans recognize named entities? We found that even without context, humans can still tell if a string is probably a named entity or not. We also assume that humans rely on context to verify NE-like candidates. With context, they could filter false candidates and find other alternations.

To implement this idea, we propose a three-phase framework: candidate generation, filtering, and recovery, as shown in Figure 3.1:

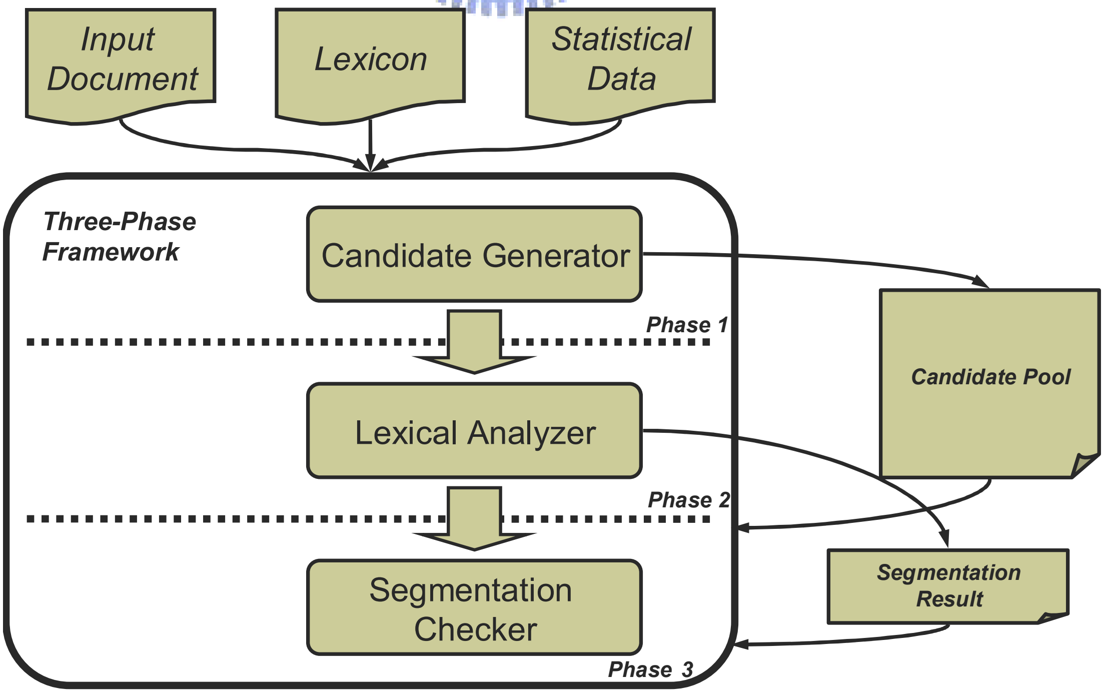


Fig. 3.1 An overview of our system.

In the first phase, all possible candidates of various kinds of named entities in the input document are extracted. Notice that this process is inevitably both over-generating and under-generating. Because of the filtering process, the candidate extracting can be tuned to have a higher recall rate and to sacrifice precision a little for a moment.

Statistical approaches are adopted in the candidate generation phase. The reason is that names are given by people. Therefore, there is no exact answer if a string is a name or not. The only thing can be judged is how likely the string is to be a name. As for computers, to estimate the likelihood of names is basically a fuzzy problem. If a character is more likely to appear in a name, it has a better fuzzy value. The detail of how fuzzy logic and statistic estimation are applied will be discussed in the next chapter.

The second phase of the system is false candidates filtering. How do we verify which candidates are true named entities and which ones are false? False candidates are either a common word or composed of fragments of common words and named entities. The first case has less impact on subsequent applications. The second case usually results ambiguous segmentations. Verification of these candidates could be viewed as an ambiguity resolution problem. If we can judge which segmentation is correct or more proper, we could also verify which candidates are true named entities. Humans usually resolve this by common senses, shallow syntactical and semantic analysis (i.e. contextual information), or deep syntactical and semantic analysis of the sentence.

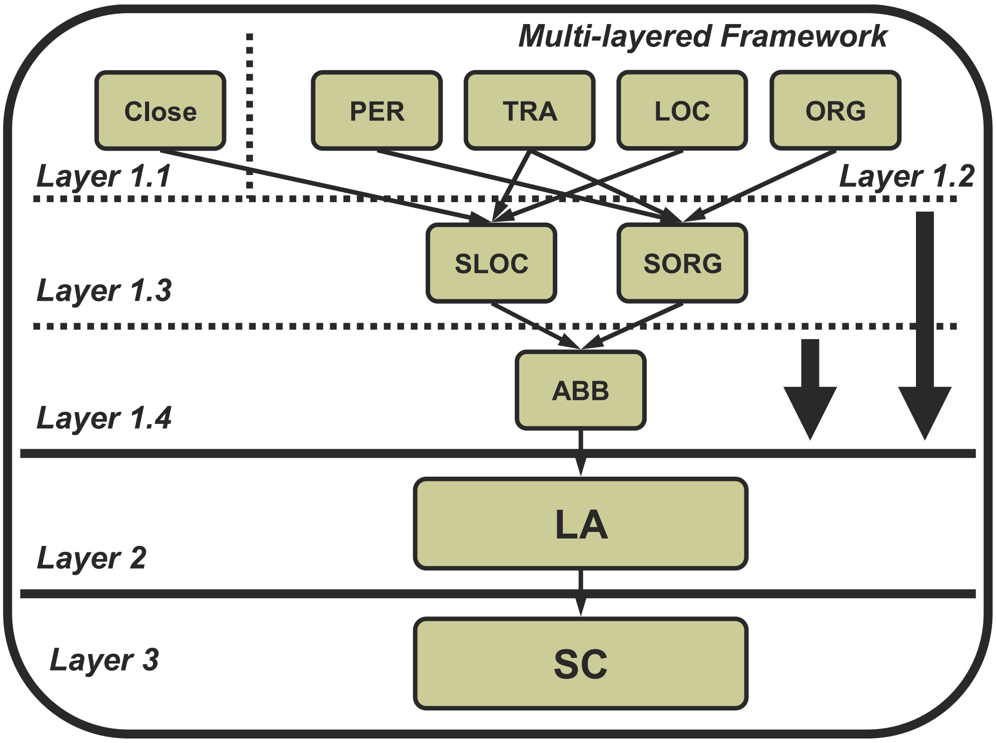
When it comes to computers, contextual information and built-in common senses database can solve a part of the problem. On the other hand, it is difficult to perform a deep analysis on a Chinese sentence. It is impractical to do deep analysis unless we could ensure the correctness of it.

Fortunately, because of the regularity of lexical choice of modern Chinese, many simple approaches of segmentation ambiguity resolution have good performances. No matter what simple methods it takes, heuristic rules or stochastic estimation, if there are no OOV words, most lexical analysis methods show great precision in ambiguity resolution as mentioned in Chapter 1. That is to say, if we got a high recall in the extraction of NE candidates, most of the segmentation ambiguities caused by false candidates are supposed to be resolved by conventional word segmentation methods. We choose a heuristic approach, which is mainly driven by maximal-matching rules, to resolve segmentation ambiguities.

The third phase of the system is the recovery. The recovery mechanism is used to revive some obviously incorrect results of the first two phases. There are two major target types to be recovered: over-segmentation caused by under-generation and under-segmentation caused by over-generation. Through the detection of these anomalies, e.g. a succession of single-character words indicating over-segmentation, part of un-extracted named entities could be revived.

3.2. A Multi-Layered Framework

The three phases of our system is further split into a multi-layered framework as depicted in Figure 3.2:



Close: Close-ended NE Model
PER: Personal Name Model
TRA: Transliteration Name Model
LOC: Location Name Model
ORG: Organization Name Model

SLOC: Whole Location-like NE Model
SORG: Whole Organization-like NE Model
ABB: Abbreviation Model
LA: Lexical Analyzer
SC: Segmentation Checker

Fig. 3.2 The multi-layered framework of our system.

In our framework, models in a same layer are independent; different layers are also independent except for the I/O relationship. The I/O interface among layer 1.1 through layer 2 is a candidate pool comprising extracted candidates, candidates' types, candidates' counts, and candidates' fuzzy values. The upper layers put their outputs into the candidate pool, and the lower ones then acquire them from the pool.

The reason to adopt a multi-layered framework is that there are various kinds of named entities, and each type of named entities has its own characteristics. It is necessary to build one specific model for each type of named entities. If we combine all models into one "big model," there will be two major problems. First, it is complicate to solve the competition among candidates generated by different models. Second, there are many different types of named entities. We cannot build all models at once. The one-big-model approach would make it difficult to add new models one by one.

Sometimes we do not want all models to be applied. The frequency and importance of different kinds of named entities in documents in different domains are varied. For example, in documents about tourism, the frequency and importance of location names are far higher than them of personal names. In this situation, if we could disable the personal name recognition model, the performance might be boosted. A layered framework could retain the flexibility of enabling or disabling some model individually according to different applications.

In our system, there is also a cascading relationship among different extraction models, i.e. they have to be applied in order sequence. The whole NE candidates are composed of genuine name candidates and specific suffixes, and the abbreviation candidates are generated from the whole NE and genuine name candidates. Figure 3.3 shows the cascading generation of candidates:

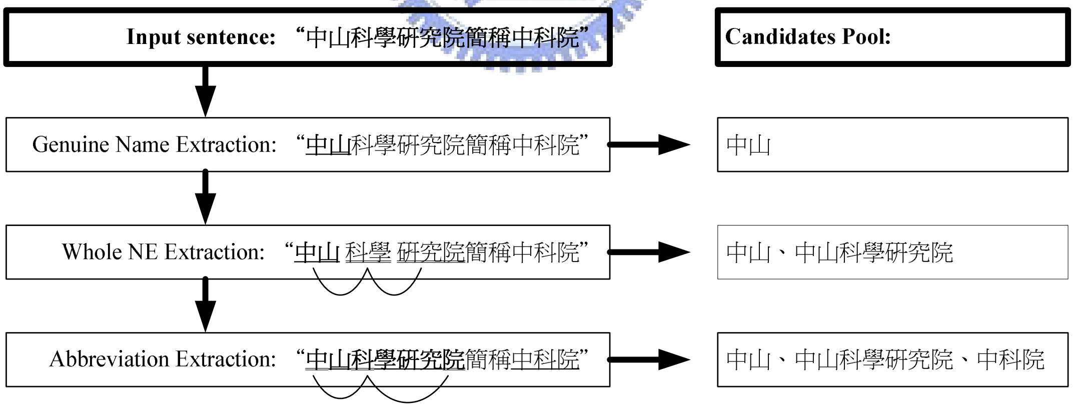

Input sentence: “中山科學研究院簡稱中科院”
Candidates Pool:
Genuine Name Extraction: “中山科學研究院簡稱中科院”
中山
Whole NE Extraction: “中山 科學 研究院簡稱中科院”
中山、中山科學研究院
Abbreviation Extraction: “中山科學研究院簡稱中科院”
中山、中山科學研究院、中科院

Fig. 3.3 The cascading generation of NE candidates.

If the cascading relationship is layered, the control and data flows of the system are more straightforward and easier to modify and maintain. It is also easier for us to add new models on a proper position of the cascade structure if it is layered.

Lastly, if the system is layered, the resolution of overlapping ambiguities among different candidates could be postponed to the next phase as a problem in general word segmentation. For example, to the sentence “孫中山是我們的國父”, “孫中山” would be recognized as a personal name, and “中山” would be recognized as a location name. If the system is not layered, an additional mechanism is needed to solve the overlapping ambiguity between these two candidates. However, in our system, both candidates are retained in the generation phase and the ambiguity will be solved by the lexical analyzer.

3.3. Data Collection

A large amount of data resources are needed for a computer to process natural languages. These include lexicon or lexical information of words, syntactical and semantic information, training data of statistic models, database of common senses, etc. A great deal of works must be paid on data collection. The quality and the selection of training data have great impact on the performance of the system.

3.3.1. Lexicon Building

As we mentioned in Chapter 1, the first step of natural language processing relies on lexicons. Because of the importance of lexicons, many previous researches develop dictionaries of their own. Thus there are many lexicons publicly available. Since the syntactical or semantic information are not necessary in our system, the completeness of word collection is our major concern.

Two lexicons are combined to achieve the goal of complete compilation in our system. The first one is How_Net 1.0 developed by Dong & Dong (2000) [14], which is free resource with bilingual lexicon of simplified Chinese and English. However there are some pre-works needed to be done before we use it. First, the English part and the description of syntax and semantics are removed, and only Chinese words are retained. Then the duplicate entries are discarded. Entries which are actually clauses, not words, are also discarded. Finally, the simplified Chinese is translated into traditional Chinese. After these steps, there are about 50,000 entries remaining. This lexicon is used solely in the probability estimation method mentioned in Chapter 4.

The second lexicon is extracted from Sinica Corpus developed by CKIP (1995) [12]. Sinica Corpus is a traditional Chinese corpus tagged with part-of-speech. All words appearing in the corpus except named entities are collected. There are about 100,000 word types in it. This corpus could help us chiefly in two aspects. First, there are Taiwanese idioms in Sinica Corpus, which do not appear in How_Net. Second, most of often seen derived words are tagged in Sinica Corpus. Derived words are generally not compiled in lexicons. The combination of these two lexicons contains about 125,000 word entries.

The basic knowledge needed by each model, like possible suffixes of location and organization names, is also compiled in the lexicon. Besides the collection of words, we also need a tag to encode their features. A 16-bit tag, or an integer, is used to denote the type of the entry. Each bit with a true value indicates that this entry is a member of the corresponding type. The meaning of these bits is listed in Table 3.1:

Table 3.1 The meaning of 16-bit lexicon tags.

bit 0	bit 1	bit 2	Bit 3	bit 4	bit 5	bit 6	bit 7	bit 8~15
Common Word	Place Name	National or Area Name	Location-like Suffix	Business Type	Affair Type	Organization-like Suffix	Quantitative Unit	Reserved Fields

Take an example, “里” is a common word, a location-like suffix, and a quantitative unit, so the value of its tag is (2^0 + 2^3 + 2^7 = 137). The reason that we need these fields for each individual model will be discussed in the next chapter.

3.3.2. Training Data

Statistic models are based on training data. Quality of training data also influences the quality of statistic models. Since there are no ready-made training data, samples we use are all collected from the Web.

There are three pieces of data used in the personal name model. The first one is the statistics of the number of families for each surname in Xiamen (廈門) city [45]. The second one is a collection of about 20,000 personal names in Taiwan, which is randomly collected from the Internet. The third one is the name list of examinees of Taiwan's Joint College Entrance Exam in 2003.

The reason to use the statistics of surnames in Shamen is that there are no ready-made data about surnames in Taiwan available on the Internet and the distribution of surnames in Shamen is similar to that in Taiwan. The second piece of data is used in the training of the given name model and the third one is used to complement rare surnames and adjust the statistical bias of them.

The reason why we used the second piece of data instead of the third one in the training of given names is that the style of naming is greatly affected by the fashion of the times. Names of some specific ages cannot represent the naming distribution of whole population. For example, almost no teenagers currently in Taiwan are called “金塗” or “罔市”.

As to transliteration names, about 20,000 transliteration names are collected, which include names of all country in the world, major cities and places, historic personages, and others.

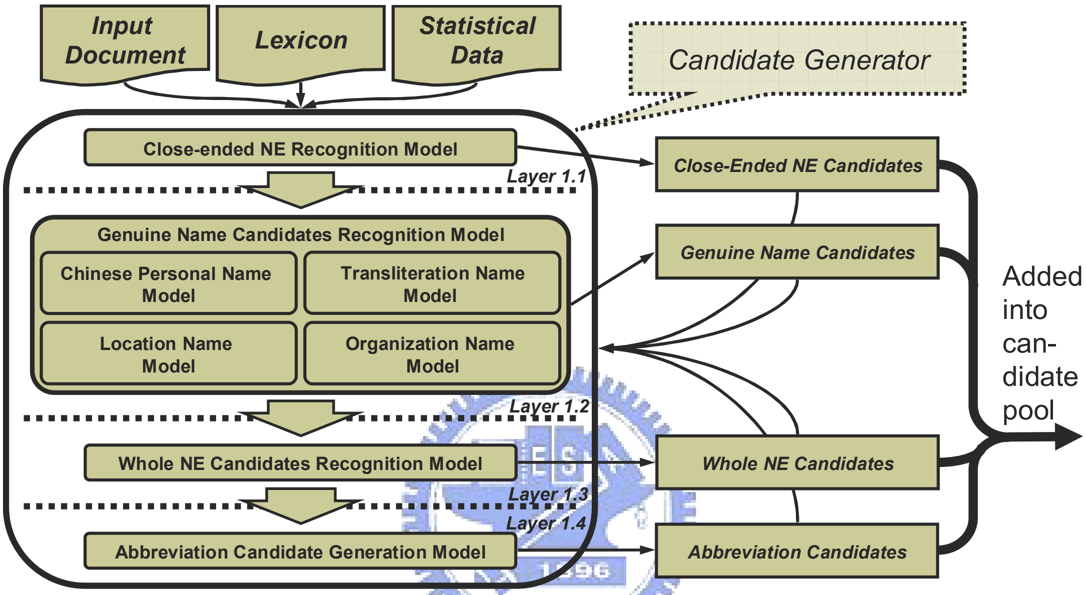
Names of administrative division at town level (鎮) in Mainland China and division at village level (村) in Taiwan, totally about 20,000 names, are used to train the location name model.

There are two steps in the collection of organization names. In the first step, names of all listed companies and companies registered in a portal site are collected as seeds of gathering more training samples and to obtain three kinds of organization suffixes (business types, affair types, and organization types). These seeds are then used to acquire company names from the web site of 經濟部商業司. We collected about 100,000 company names as the real training sample.

Other necessary data, including location-like-suffixes, quantitative units, and most organization-like suffixes, are all handcrafted with dictionaries.

Chapter 4 Named Entity Candidate Generation

The candidate generator is used to extract all possible named entity candidates in input documents. There are four layers in the candidate generator to handle four sorts of NEs: close-ended NEs, genuine names, whole NEs, and abbreviations.



Input Document
Lexicon
Statistical Data
Candidate Generator
Close-ended NE Recognition Model
Layer 1.1
Genuine Name Candidates Recognition Model
Chinese Personal Name Model
Transliteration Name Model
Location Name Model
Organization Name Model
Layer 1.2
Whole NE Candidates Recognition Model
Layer 1.3
Layer 1.4
Abbreviation Candidate Generation Model
Close-Ended NE Candidates
Genuine Name Candidates
Whole NE Candidates
Abbreviation Candidates
Added into can-didate pool

Fig. 4.1 The overview of the candidate generator.

The first layer is the close-ended named entity recognition model. Numerical expressions in input documents are extracted here. Since the extraction of close-ended named entities is not the focus of this paper, and previous researches [22] have solved this problem very well, a single simplified rule is applied to recognize most of close-ended named entities in our system. The rule we use is as follows:

$$[\text{“第”}] + (Numerals)^{+} + [Qualifier] + [Unit]$$

The set of numerals also comprises decimal point “.” and percent symbol “%” besides 0 to 9. Time expressions could also be covered by the above rule, where time units like “月”, “日”, and so on, are viewed as a sort of quantitative units. Therefore, “1999 年 3 月 4 日” will be treated as “1999 年 / 3 月 / 4 日”. This simple rule cannot cover all close-ended NEs, of course. However, our focus is open-ended ones. The purpose of this rule is just to prevent unrecognized close-ended NEs affect the performance of the recognition of open-ended ones.

The rest three layers are all responsible for the recognition of open-ended NEs. The second layer mainly takes statistical approaches to recognize genuine names. There are many different types of named entities. Many sorts of them rarely appear in the document. Therefore, these names are also less important. It is not worth to build models for each type of these names. First, the effort we paid is not proportioned to the effect we got. Second, because the appearing frequency of these names is very low, adopting specific models for them might not improve total performances lot.

However, suffixes are strong features. They are easier to be recognized, and chances of error recognition are comparatively low. Therefore, a compromised method is adopted that only models for four kinds of genuine names are implemented at present in our system. They are personal names, transliteration names, location names, and organization names. These four kinds of genuine name candidates would be used to form various types of NEs with corresponding suffixes in the third layer. For instance, if a personal name candidate is followed by a publication suffix, they will be recognized as a whole publication name, like:

"余光中"(personal name) + "詩選"(publication suffix)

→ *"余光中詩選"(publication name)*

In the third layer, for the same reason above, all NE suffixes are roughly classified into three categories: ones with similar corresponding genuine name types to location suffixes, ones with similar corresponding genuine name types to organization suffixes, and others. The first category covers all location names, racial names, etc. The second one comprises all organization names except for racial names, facility names, publication names, part of miscellaneous names, etc. The third one includes feat names, culture names, and so on. Among these three categories, only the first two are addressed by our system. These two categories are called "*location-like NE*" and "*organization-like NE*". Names belonging to the same category will be addressed by the same corresponding model. There are two main advantages following this way. First, the times spent on designing models and collecting data is saved. Second, confidence brought by suffixes could alleviate the deviation on statistics brought by a compromised approach.

Named entities extracted in the above two layers are used to find possible abbreviations in documents in the fourth layer.

The generic statistical estimation methods for genuine name models are discussed in Section 4.1 and genuine name models are described in Section 4.2~4.5. Whole NE models are addressed in Section 4.6~4.7. Section 4.8 states the abbreviation model.

4.1. Fuzzy Logic and Statistic Estimation

As we mentioned before, the recognition of *genuine names* is basically a fuzzy decision problem to computers. There is no exact right or wrong answer for a string to be a name. The only problem is how likely it is. Fuzzy values represent strings' likelihood or properness to be a name. Since Chinese is a character-based language, methods of estimating fuzzy values are generally also character-based. Names are composed of several characters. There are several ways to transform the member characters' fuzzy value to the string's fuzzy value.

If we try to imitate humans' thought, the first idea might be neural networks. Regardless of the network adopted, multi-layer perceptron or support vector machine, it is actually a fantastic idea. However, in natural language processing or named entity recognition, neural networks have their inherent drawbacks.

To use neural networks, we must encode Chinese characters first. The frequently used Chinese code is Big-5. Big-5 is encoded according to the number of strokes and the order of radicals. There are few semantics and syntax information involved in the encoding of Big-5. Therefore, two characters with similar Big-5 codes might be very different in their properties. If we use Big-5 code for training directly, many layers and percetrons are supposed to be needed. It is hard to design such networks, and the training time and performance might not be satisfying. Another problem caused by using Big-5 code directly is the data-sparseness problem. While a similar encoding does not mean a similar meaning, the properties of characters that do not appear in the training data are very likely to be wrongly judged by the network.

Because of the problem with neural networks, stochastic language models are usually adopted to estimate the likelihood of a candidate to be a named entity. The fundamental principle is that the string with a higher probability or frequency to be a name has a higher fuzzy value or likelihood. There are several ways to estimate the fuzzy value of a string from the statistic data based on characters. These models include Markov models, bi-gram models, unigram models, etc.

Each model has its advantages and disadvantages. Generally speaking, more complex the model is, more precisely it estimate, and more training data it needs. Besides that, the data-sparseness problem is more likely to happen. The amounts of features of different types of named entities are varied, so each type has its own best-fit model. In this paper, to simplify data collecting and training, the unigram model is adopted. Additionally, some supplementary information such as positional feature is exploited to support statistical models.

Generally there are two major ways to estimate the fuzzy value of a single character. The first one uses frequencies, i.e. characters' counts in named entities in the training corpus:

$$freq(typ|c)=counts(typ, c)$$

The other one is using the probability, i.e. characters' counts in named entities dividing by the total counts in the training corpus:

$$prob(typ|c)=counts(typ, c)/counts(c)=freq(c)/counts(c)$$

For example, if the character “陳” appears for 2,000 times in the training corpus, and among these 2,000 times it appears as a surname for 500 times, then the frequency of “陳” as surnames will be 500 and the probability equals to 500/2000=0.25 on the other hand. Both frequencies and probabilities have their advantages and disadvantages.

Frequencies stand for differences among naming-characters. They represent popularities of characters to be used in names of some type. If some character is used in more names, it has a higher frequency. For example, since “俊” is used in more given names than “昱”, “俊” has a higher frequency being given names than “昱”. If frequencies are used as fuzzy values, a higher recall will be obtained with common names like “陳俊明” and “林美慧”, because common names will have a higher fuzzy value through this way.

Probabilities stand for differences among all characters. They represent possibilities of characters to be used in a name of some type. If some character appears more frequently in names than in common words, it has a higher probability. For example, since “昱” is very rarely used in common words and on the contrary “俊” is an often seen character in common words, “昱” has a higher probability being given names than “俊”. If probabilities are used as fuzzy values, a higher precision and a higher recall will be obtained with rare names like “昝家驤” and “班婕妤”. However, it has a lower recall with common names comparing with using frequency. Table 4.1 shows several normalized frequencies and approximate probabilities of several characters being given names as examples to explain the above statements:

Table 4.1 Normalized frequencies and approximate probabilities of several characters being given names.

Character	**Common naming character?**	**Common character?**	**Normalized Frequency**	**Approximate Probability**
俊	YES	YES	0.85	0.61
昱	YES	NO	0.68	>0.99
門	NO	YES	0.10	<0.01
媜	NO	NO	0.10	>0.99

Above frequencies are computed from our training data, which would be detailed in following paragraphs, and these probabilities are estimated from part of Sinica Corpus.

A hybrid statistics is adopted in our system to take advantages of both frequencies and probabilities. With common naming-characters, frequencies are adopted to get a higher recall rate with common names. With rare naming-characters, a probability model is adopted to complement frequencies' insufficiency with rare names. The resulting model looks like:

$$\ell\ (typ|c) = Max\{\ freq(typ|c), prob(typ|c)\ \}$$

The major difficulty with using probabilities is that it is hard to estimate the probability of some character to be used in a name. Comparing with frequencies, the estimation of probabilities needs a real training corpus. It is easy to collect ten thousands or hundred thousands names, but much harder to acquire a training corpus which contains ten thousands or hundred thousands names. Even if we successfully obtain a huge corpus with hundred thousands names, it is difficult to ensure whether the corpus is balanced. The density of names in documents is greatly varied in different domains. The sampling of training corpus also affects the statistic result very much. Even if we can ensure the balanced sampling of corpus, the locality problem still cannot be prevented. While a name appears in a document, it is very likely for this name to reappear in the same document. This causes the probability of some rare-used characters to be over-estimated.

Because of the difficulty in estimating probability mentioned above, we propose to use inverse common frequency model to approximate the probability model:

$$icf(c)=1/(freq(common\ word|c)+1)=1/(counts(common\ word,\ c)+1)$$

Probability models are mainly used to estimate the probability of rarely seen events. To rarely seen events, usually:

$$counts(common\ words,\ c) \approx counts(\sim typ,\ c)$$

And:

$$counts(typ,\ c) \leq 2$$

In this case, $icf(c)$ is approximate to $prob(c)$:

$$prob(c) = counts(typ,c)/(counts(typ,c)+counts(\sim typ,c))\ where\ counts(typ,c) \leq 2$$

$$\approx 1/(counts(\sim typ,c) + 1) \approx icf(c)$$

Further, we assume that $counts(common\ word,\ c)$ is in direct proportion to the number of lexicon entries in which the character c appears, i.e. morphemes that are more frequently used in word-formation are also more frequently used in real documents. Under these assumptions, we use inverse lexicon counts to approximate the probability model:

$$ilc(c) = 1/(Num\_of\_Lex\_Entries(c)+1) \approx icf(c) \approx prob(typ|c)$$

For example, if we have four lexical entries contain the character “上”, such as {上, ADV}, {上, V}, {上台, V} and {上路, V}, then $prob(typ|上)=1/(4+1)=0.2.$ Because $prob(typ|c)$ is ranged from 0 to 1, $freq(typ|c)$ also needs to be normalized to 0 to 1. The distribution of raw data of $freq(typ|c)$ is conformed to Zipf’s Law [45], that:

> $P_n \approx 1/n^a$*, where* P_n *is the frequency of occurrence of the* n^{th} *ranked item and a is close to 1.*

Values with often seen characters are too high and the distinctions among low frequency characters are not wide enough.

Therefore, the logarithm function is taken on the raw data to smooth the distribution curve, and then the result is normalized to 0.1 to 1.

$$freq^*(typ \mid c) = \underset{0.1,1}{Norm}(\log(freq(typ \mid c)))$$

Notice that the lower bound of *freq(typ|c)* is set to 0.1, not 0. This is because the meaning of events that appear once is greatly different from the meaning of unseen events.

Our final character likelihood model looks like:

$$\mathcal{L}\ (typ|c) = Max\{\,freq^*(typ|c),\ ilc(c)\ \}$$

4.2. Personal Name Model

The personal name model is responsible for the generation of personal name candidates. In this paper, we will only focus on the recognition of formal Chinese personal names. The same principle could be adapted to other formal hanzi names in different countries and dialects such as Japanese and Korean simply by changing the set of surnames and statistic data that represent the preference of given names.

A formal personal name is composed of *surname* + *given name*, and there are usually appellations or human-subject verbs following it. Therefore, there are three major features to recognize formal personal names:

- Surname (e.g. “陳”, “王”, “劉”)
- Given name (e.g. “志明”, “淑芬, “俊宏”)
- Appellation and human-subject verb (e.g. “先生”, “表示”, “來”)

There are more than 3,000 Chinese surnames, but most frequently used 500 surnames cover more than 99% of populations [47]. Preferred characters that would be used in given names are limited. The set of appellations and human-subject verbs is also limited.

To simplify our model, only first two features are exploited in this paper, and we assume that choices of surnames and characters of given names are independent. Therefore, a unigram model is adopted to estimate the likelihood of a string to be a personal name: (Capitalizations are used to discriminate the likelihood of a string from that of a character.)

$$\mathcal{L}(PER|s) = ArgMax\ \{\mathcal{L}(SUR|s1) * \mathcal{L}(GIV|s2)\}\ \textit{for every substring s1 and s2, where}$$

$$s = s1 \bullet s2,\ \text{“} \bullet \text{”}\ \textit{denotes the string concatenation}$$

Chinese surnames are either one or two-character long. Sometimes married women would add their husband's surname in front of their surname. However, things are slightly different with people with a plural surname. First, there are few people with a plural surname. Second, these people usually would not add their husband's surname in front of their surname. Therefore, we restrict the combinations of Chinese surnames in three:

1. single surname
2. plural surname
3. two single surnames

The likelihood of surname is defined as:

$$\mathcal{L}(SUR|s) = \begin{cases} \ell(SUR|c1) & \textit{when s is constituted of one character} \\ Max\{Avg(\ell(SUR|c1), \ell(SUR|c2)), \ell(SUR|c1c2)\} & \textit{when s is constituted of two characters} \\ 0 & \textit{when s is longer than two characters} \end{cases}$$

where *$\ell(SUR|c)$* is a little dissimilar to other character likelihood functions. Because we use inverse lexicon counts to approximate probabilities, every character has a non-zero probability. However, surnames are not arbitrarily given. Probabilities of some character to be surnames are actually zero. With these characters, the original estimation method might cause unnecessary over-generations. To prevent this problem, only surnames appearing in our training data are adopted as correct surnames. The estimating methods mentioned in the previous section are applied on these surnames. Other surnames are assigned a likelihood of zero.

There is another consideration for the robustness of the model. Unseen surnames are supposed to be assigned a likelihood lower than one, and the recovering mechanism still has chance to recognize it. However, practically there is very little difference between assigning a zero and a low likelihood, so the zero likelihood is assigned in this paper.

Chinese given names are either one character or two characters long. Therefore, $\mathcal{L}(GIV|s)$ is defined as:

$$\mathcal{L}(GIV|s) = \begin{cases} \mathcal{L}(GIV|c1) & \textit{when s is constituted of one character} \\ Avg(\mathcal{L}(GIV|c1), \mathcal{L}(GIV|c2)) & \textit{when s is constituted of two characters} \\ 0 & \textit{when s is longer than two characters} \end{cases}$$

Some researchers proposed to use statistics of single given name, the first character of double given names, and the last character of double given names separately [44], or exploit sexual information [5]. It needs triple training samples to take the first way, and it is not easy to collect thousands of single given names. If the second way is followed, we still need double samples. Usually the names collected are not classified by genders and it is a tedious task to classify them.

In fact, through our experiment, the precisions that might be gained by separating first character from second character or male from female are very limited. Nearly no false positive errors with personal names in our experimental results could be solved by exploiting positional or sexual information. That is because there are generally no explicit positional or sexual features with the frequently used characters that are often wrongly recognized as a part of personal names. Therefore, the simplest statistical counts are taken in this paper. Part of $\mathcal{L}(SUR|c)$ and $\mathcal{L}(GIV|c)$ are listed in Appendix B.

Notice that no matter with surnames or given names, the average of likelihoods of characters is taken. The geometric mean function, i.e. $avg(a, b) = (a*b)^{0.5}$, is applied to compute the average. Arithmetic means, i.e. $avg(a, b) = (a+b)/2$, emphasize too much on frequently used characters. As Fig. 4.2 shows, if one of two characters is a frequently used character, the likelihood of this string would easily exceed the threshold we set. It is contrary to the result we hope for. We hope that, if one of two is a very rarely used character, the likelihood of the string is supposed to be lower than our threshold. Geometric means are more closed to this target than arithmetic means, and harmonic means, i.e. $avg(a, b) =2ab/(a+b)$, emphasize too much on rarely seen characters. For example, “李明沒帶錢去李大同家”, obviously “大同” is much more like a given name than “明沒”. We know that *ℓ(GIV|明)*=0.938*,* *ℓ(GIV|沒)*=0.012*,* *ℓ(GIV|大)*=0.553*,* and *ℓ(GIV|同)*=0.324. If arithmetic means are adopted, *ℒ(GIV|明沒)* = (0.938+0.012)/2=0.475 > *ℒ(GIV|大同)* = (0.553+0.324)/2 = 0.439. However if geometric means are adopted, *ℒ(GIV|明沒)* = $(0.938*0.012)^{0.5}$ = 0.110 < *ℒ(GIV|大同)* = $(0.553*0.324)^{0.5}$ = 0.423.

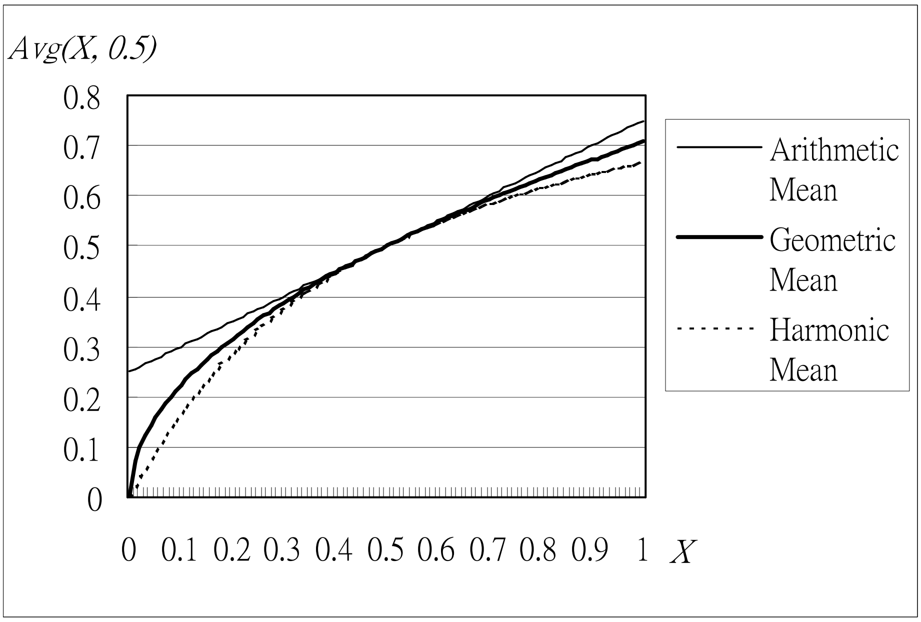


Fig. 4.2 Curves of three different mean function of *Avg(x,0.5)*.

The functions mentioned above infer that the value of $\mathcal{L}(PER|s)$ is between 0 and 1. In fact, most of $\mathcal{L}(PER|s)$ of often seen personal names are between 0.4 and 0.8, and $\mathcal{L}(PER|s)$ of strings which are not Chinese personal names is usually under 0.3.

Table 4.2 $\mathcal{L}(PER|s)$ of several strings.

s	$\mathcal{L}(PER|s)$
李登輝	0.5325
陳致遠	0.5328
張愛玲	0.6425
張惠妹	0.6425
戴高樂	0.2774
柯林頓	0.1045
張開眼	0.1612
英文書	0.0811
路人甲	0.1351
木頭人	0.0472

A simple experiment is designed to decide the threshold of $\mathcal{L}(PER|s)$ between personal name candidates and rest ones. $\mathcal{L}(PER|s)$ of personal names of all training data and all possible fragments of every entry in lexicon are computed first. Then we estimate what threshold value should be to maximize the *weighted f-measure* of the recall rate of the training data and the excluding rate of lexicon entries. Weighted f-measure is proposed by van Rijsbergen (1979) [34]:

$$F_b = \frac{(b^2+1)ER}{b^2E+R}$$

E = excluding rate = *# of false negative / # of false*

R = recall rate = *# of true positive / # of true*

Because appearing frequencies and importance of personal names in documents are different from them of common words, and we do not have a large corpus tagged with named entity types to adjust type and token frequency difference, the weight b of f-measure function is set to 0.5, i.e. common words are twice important than personal names. The figure of the distribution of $\mathcal{L}(PER|s)$ and the curve of weighted f-measure $F_{0.5}$ is showed below:

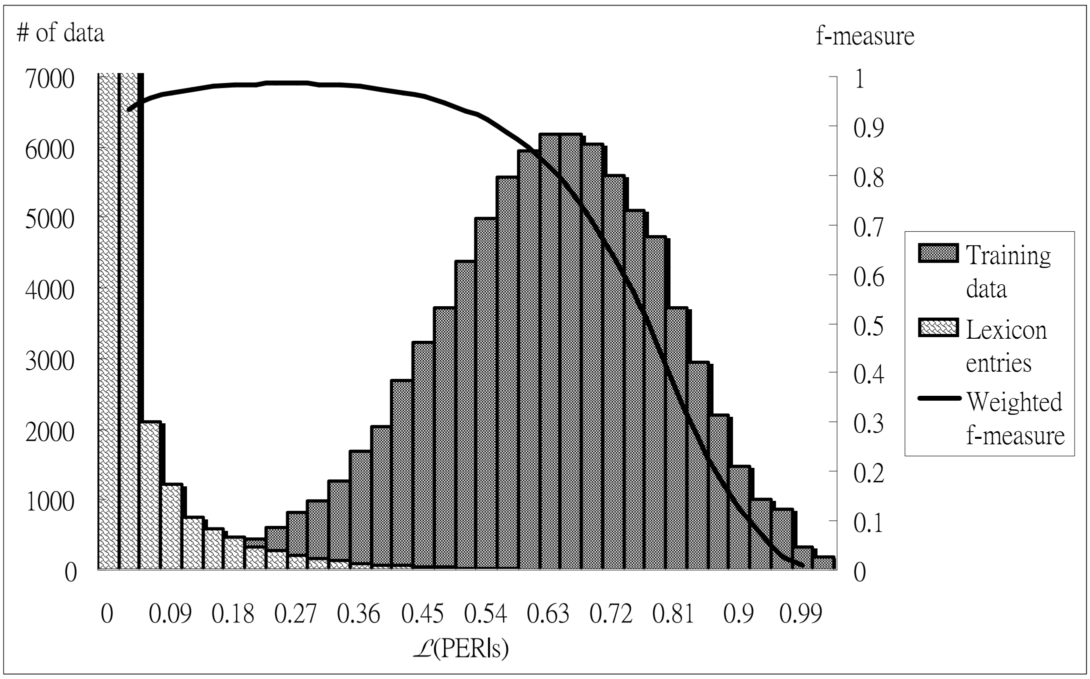


Fig. 4.3 The distribution of $\mathcal{L}(PER|s)$ and the curve of weighted f-measure $F_{0.5}$.

According to experimental results, the maximal f-measure is located at the threshold value 0.26. Therefore, we set the threshold of personal name model to 0.26. Every string with a $\mathcal{L}(PER|s)$ over 0.26 will be treated as personal name candidates and going to be filtered by the lexical analyzer. To address the true ones which have its $\mathcal{L}(PER|s)$ under 0.26, a remedial method is applied besides the recovering mechanism. The reoccurrence in the document is employed to boost $\mathcal{L}(PER|s)$ of the string. The following function is adopted to make use of reoccurrence:

$$\mathcal{L}^*(PER|s)=\mathcal{L}(PER|s) * k^{Reoccurrence(S)}$$

$$k=\begin{cases} 2 & \textit{when the length of the input document} \\ & \textit{is less than 400 characters} \\ 1+400/LEN(Document) & \textit{elsewhere} \end{cases}$$

If the $\mathcal{L}(PER|s)$ after boosting is over 0.26, this string will be added into personal name candidates. Notice that k is decreased when the length of input document is longer than 400 characters to prevent reoccurrences of frequently used words from causing false detecting. This function is also applied on other kinds of named entities.

4.3. Transliteration Name Model

The transliteration name model is responsible for recognizing transliterated *genuine names*. As we mentioned in Chapter 2, there are three types of translated names. This model only addresses the phonetically translated, or transliterated, ones. The Chinese-style ones could be solved by a similar model to personal name model, and the translating in meaning ones are a much harder task to overcome and it is beyond the scope of this thesis.

Phonetic type of transliteration names is usually composed of sequences of meaningless characters. Therefore, there are no obvious morphological features in the internal structure of transliteration names, and the variation in length of them is relatively large. (Comparatively speaking, other *genuine names* are usually constituted of one to four characters; however a transliteration name may be as long as nine characters, such as “艾力克斯羅德里奎茲(Alex Rodriguez)”.)

To simplify our model, we assume that likelihoods of component characters of transliteration names are independent to one another. Thus a unigram model is used to recognize transliteration names:

$$\mathcal{L}(TRA|s) = \underset{k=1\ldots n}{Avg}(\mathcal{L}(TRA|ck))\ where\ S=c1\ldots cn$$

Since the number of characters that will be used in a transliteration name is limited, $\mathcal{L}(TRA|c)$ is estimated in a similar way like surnames. First, all characters which are possible to appear in a transliteration name are sieved out by hand. Then $\mathcal{L}(TRA|c)$ where c is outside of this set is set to zero. Characters belonging to this set but not appearing in training data are given a low $\mathcal{L}(TRA|c)$. The general estimating method is adopted to compute $\mathcal{L}(TRA|c)$ of characters appearing in the training data.

Harmonic means are adopted in the transliteration name model to compute

Avg($\mathcal{L}(TRA|c)$). Because transliteration names may be very long, using arithmetic means or geometric means might cause the characters nearby long sequences with a high $\mathcal{L}(TRA|s)$ to be wrongly attached to the sequence. Therefore, the transliteration name model needs to put more emphasis on the low $\mathcal{L}(TRA|c)$ of rarely used characters than other models do. For example, if arithmetic mean or geometric mean is used on “保羅和約翰”, the common word “和” would be wrongly attached to “保羅” and “約翰” because of the high $\mathcal{L}(TRA|s)$ of them, and thus the system will incorrectly recognize “保羅和約翰” as a whole transliteration name.

Table 4.3 $\mathcal{L}(TRA|s)$ of several strings.

s	$\mathcal{L}(TRA|s)$
柯林頓	0.714
布萊德彼特	0.748
羅德里奎茲	0.733
珍妮佛洛佩茲	0.648
戴高樂	0.550
妮可基嫚	0.744
多明尼加	0.678
紐約	0.562
黃土高原	0.164
北海道	0.222

The threshold of $\mathcal{L}(TRA|s)$ is estimated in a similar way to personal names. However, the weight *b* of f-measure function is set to 1/3 instead of 1/2. That is because the frequency of transliteration names is much lower than other types of *genuine names*. The best threshold thus obtained is 0.51. The figure of the distribution of similarities and the curve of f-measures is showed below:

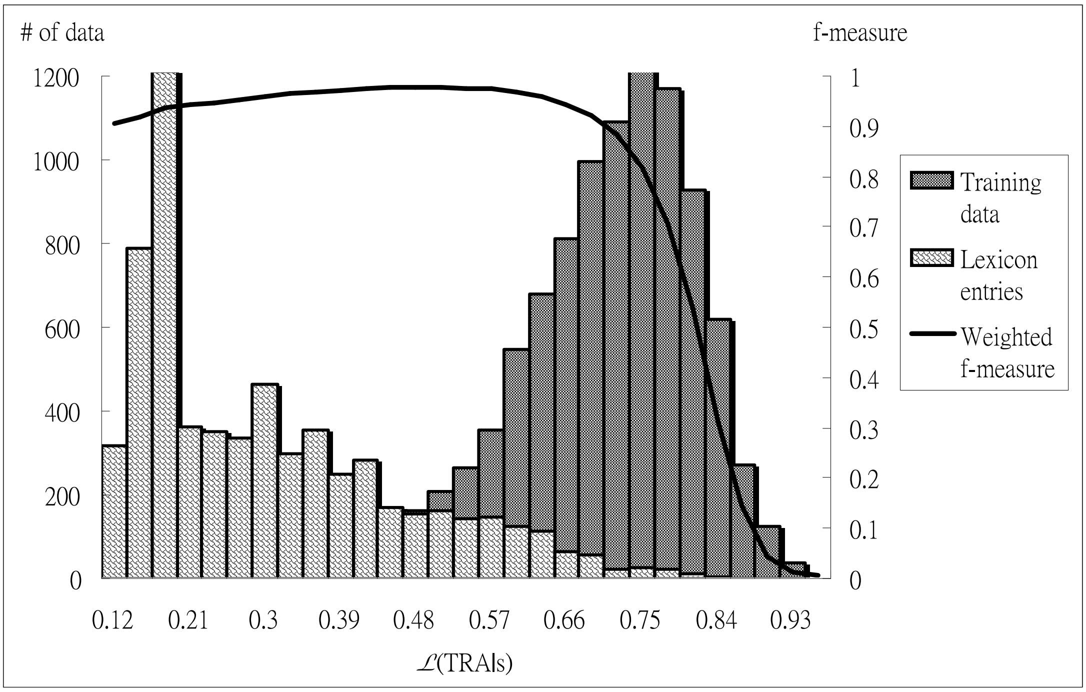


Fig. 4.4 The distribution of $\mathcal{L}(TRA|s)$ and the curve of weighted f-measure $F_{0.33}$.

Generally speaking, because the number of characters used by most transliteration names is less than 500, unigram models have little problem with recognizing the boundary of these names. However there are several common words with a high $\ell(TRA|c)$, as “斯文”, “倫理”, etc. When these words are adjacent to transliteration names, they might be wrongly recognized as part of transliteration names. This kind of situations might result lexical analyzer cannot correctly filter out the false candidate. For example, “柯林頓斯文有禮” has at least three possible segmentation:

1. 柯林 頓 斯文 有禮
2. 柯林頓 斯文 有禮
3. 柯林頓斯文 有禮

Case 1 will be filtered out correctly by the lexical analyzer driven by maximal-matching rule, however, case 3 might be wrongly chosen because the common word “斯文” is appended by the transliteration name “柯林頓”. In order to solve this problem, a filtering process must be done first in the transliteration name model.

There are two possible ways to follow. The first one is to detect the appearance of common words in the two ends of transliteration names and remove them. This is a simpler way, but it is not rational in practical. Because, in fact, “柯林頓斯文 有禮” is a correct segmentation syntactic or semantic wise. We consider that the case 2 is the correct one just because “柯林頓” is a familiar name to us.

Computers do not have common senses, so the reoccurrence feature must be exploited to handle this vague situation. Therefore, a concept called “*team*”, which is based on the reoccurrence feature, is introduced to solve the attaching problem. Basically, all substrings of possible transliteration name candidates with a length more than 1 are also possible candidates. Hence all transliteration name candidates can be grouped into *teams* according to their longest common superstring candidate. For example, “麥可”, “可喬”, “喬丹”, “麥可喬”, “可喬丹”, and “麥可喬丹” have a longest common superstring candidate, “麥可喬丹”, so they belong to the same *team* and “麥可喬丹” is called the “*leader*” of the *team*. If all appearance times of candidates are marked up, a *team* can be represented as:

$$T_{\text{leader=麥可喬丹}} = \{\text{麥可(5), 可喬(5), 喬丹(6), 麥可喬(4), 可喬丹(5), 麥可喬丹(4)}\}$$

Notice that the appearance times of superstrings are inevitably less than that of substrings.

The following algorithm is then applied:

1. Subtract leader’s appearance times from each team member
2. If the *leader* could be split into candidates with non-zero appearance times after subtraction and multisyllabic common words or frequently used monosyllabic words, discard the *leader* and members whose appearance times being subtracted to zero
3. Form new teams comprised of remaining candidates with new leaders
4. Repeat step 1~4, until no candidates could be discarded

For the above example, after step 1, the result will be:

{麥可(1), 可喬(1), 喬丹(2), 麥可喬(0), 可喬丹(1), 麥可喬丹(0)}

Then step 2 is followed. Since “麥可” & “喬丹” can form “麥可喬丹”, “麥可喬丹” and “麥可喬” will not be kept in the candidate list. The remaining candidates would be:

{麥可(1), 可喬(1), 喬丹(2), 可喬丹(1)} = $T_{\text{leader=麥可}} \cup T_{\text{leader=可喬丹}}$

Step 1 is repeated with the team $T_{\text{leader=可喬丹}}$. After subtract the leader’s appearance times, the result will be {喬丹(1), 可喬丹(0)}. The character “可” is looked up in the lexicon. Because “可” is a frequently used monosyllabic word, “可喬丹” will be discarded. Thus, the remaining candidates will be {麥可, 喬丹}, that is just the result we want.

4.4. Location Name Model

The location name model is mainly used to extract *genuine names* of locations from documents. Location names are basically a more closed set than other kinds of named entities. Most of often seen location names could be collected. As we mentioned in Section 3.3, a lot of Chinese administrative division names and foreign national names are collected in our system. Most foreign location names can be handled by transliteration name model, but there are several foreign location names not translated in phonetic spellings, like “費城” and “舊金山”. Because these names are usually named in Chinese style, many of them can be correctly recognized by the location name model.

Most Chinese location names are constituted of one to four characters. With genuine location names constituted of one character or four characters, only the collected ones or the whole NEs with suffixes would be identified. Hence our location name model only recognizes location names of two or three characters. Since the statistical distributions of characters in different position of location names are greatly varied and many of commonly naming characters of locations are also common nouns, a different way from above two models is taken to build the recognition model. The statistical data of leading characters and following characters of location names are estimated separately. Therefore a unigram model with position dependency is adopted:

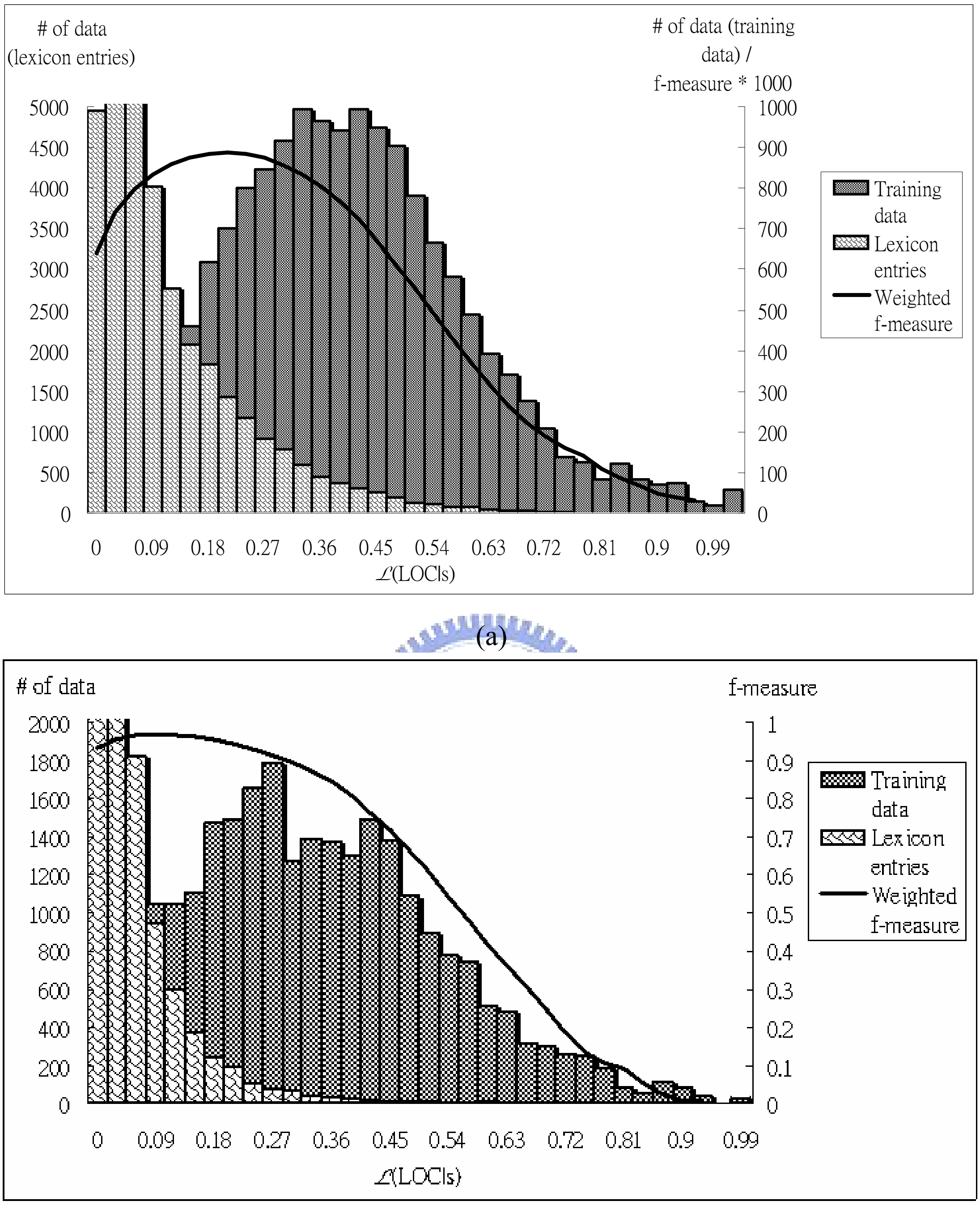
$$\mathcal{L}(LOC|s) = \begin{cases} \mathcal{L}(LOCL|c1) * \mathcal{L}(LOCF|c2) & when\ s = c1c2 \\ \mathcal{L}(LOCL|c1) * \mathcal{L}(LOCF|c2) * \mathcal{L}(LOCF|c3) & when\ s = c1c2c3 \\ 0 & elsewhere \end{cases}$$

During training, monosyllabic location names are viewed as leading characters, and the component characters of four-character-long location names are counted as leading, following, leading, and following respectively. The generic character likelihood function is adopted to estimate both $\mathcal{L}(LOCL|c)$ and $\mathcal{L}(LOCF|c)$.

Table 4.4 $\mathcal{L}(LOC|s)$ of several strings.

s	$\mathcal{L}(LOC\|s)$
台北	0.3882
高雄	0.2596
埔頂	0.3877
東湖	0.7704
重慶	0.3045
上海	0.5729
江南	0.6094
浦東	0.4013
江子翠	0.2549
風陵渡	0.1496
舊金山	0.2724
水牛城	0.1908

Since the positional information is used and thus causes the computing functions of two- and three-character-long names to be diverse, different thresholds are supposed to be set in different cases. Experiments similar to what above two models have done are applied to two- and three-character-long location names. The weight of f-measure function is set to 1/2 and 1/3 respectively. The best thresholds thus estimate are 0.24 and 0.14. The resulting statistical distributions and f-measure curves are figured below.



(a)

(b)

Fig. 4.5 (a) The distribution of $\mathcal{L}(LOC|s)$ where s is two-character long and the curve of weighted f-measure $F_{0.5}$. (b) The distribution of $\mathcal{L}(LOC|s)$ where s is three-character long and the curve of weighted f-measure $F_{0.33}$.

4.5. Organization Name Model

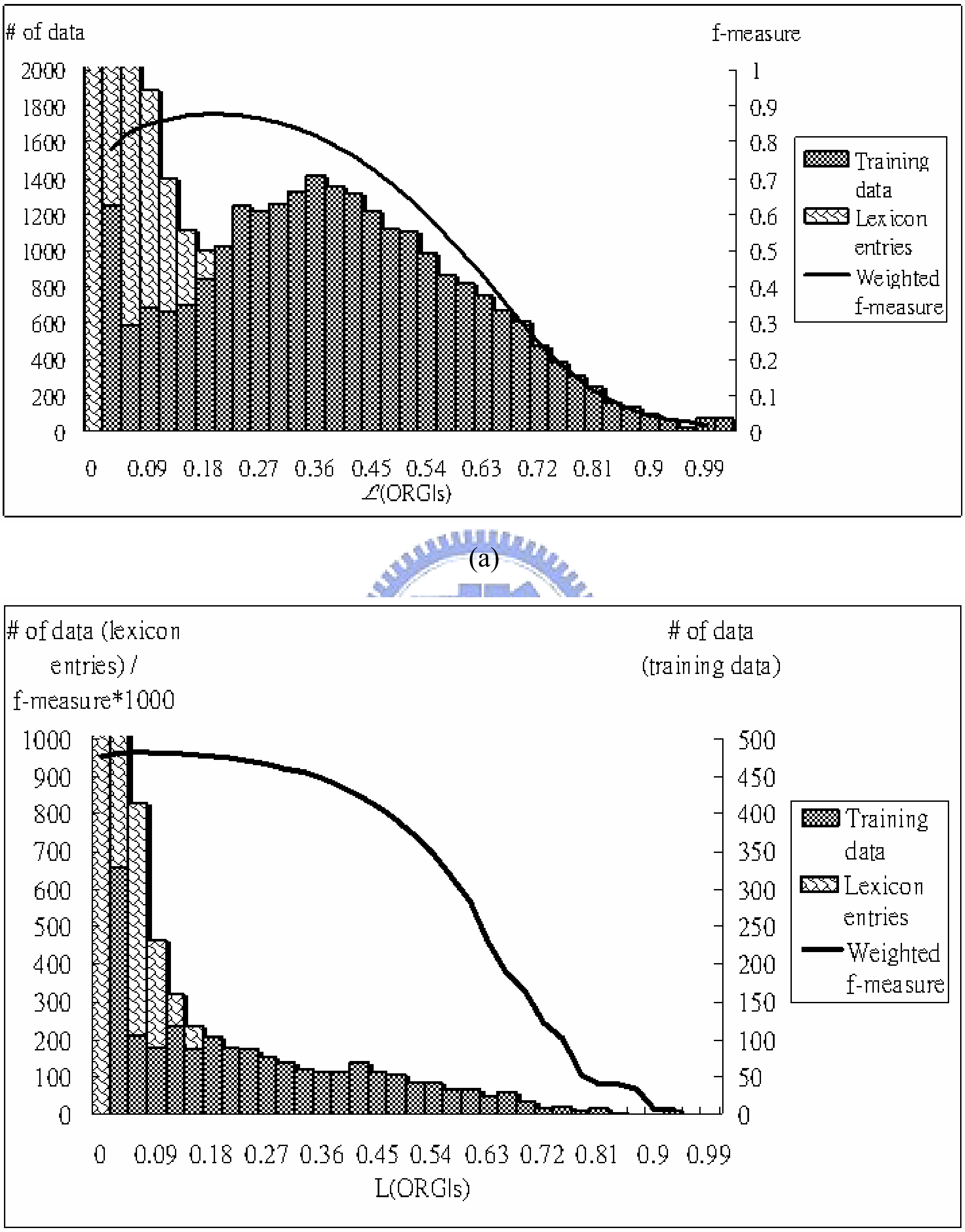
The organization name model is mainly responsible for the recognition of various kinds of genuine names of organizations. Besides these, genuine names of several named entities of title and miscellaneous type with suffixes are also the target of the model. Hence this model is more precisely called "*organization-like genuine name model*", as we have mentioned before.

Among the four types of *genuine names* that would be dealt by our system, the recognition of organization names is the hardest one. The problem is that there is nearly no internal structure within a genuine Chinese organization name, and the statistical feature is not obvious, too. Even worse, the most frequently used naming characters of organizations are also the ones that constitute the most frequently used common words. (In fact, it is very usual for common words to be an organization name.)

Except for racial names, organization names usually comprise two to four characters. (The name part of racial names is classified into location-like name in recognition instead.) Four-character-long organization names are often composed of two names or a phrase; to recognize the former, a downward layer using simple rule-driven methods is more suitable; to the latter, the performance of recognizing them with statistical model is not satisfied. Therefore, similar to location name model, organization name model only focus on names constituted of two or three characters. The positional information is also used to assist the recognition of organization names. Thus, the organization model is just a little bit different from location name model in the exploiting of positional information with three-character long strings:

$$\mathcal{L}(ORG|s) = \begin{cases} \ell(ORGL|c1) * \ell(ORGF|c2) & when\ s = c1c2 \\ \ell(ORGL|c1) * \ell ORGL|c2) * \ell(ORGF|c3) & when\ s=c1c2c3 \\ 0 & elsewhere \end{cases}$$

The generic character likelihood function is also adopted to estimate $\ell(ORGL|c1)$ and $\ell(ORGF|c2)$. Through a similar experiment to above models with f-measure weight 1/2 and 1/3 respectively, the thresholds of two- and three- character-long organization names are set to 0.2 and 0.1. The statistical distributions and f-measure curves are drawn below:



(a)

(b)

Fig. 4.6 (a) The distribution of $\mathcal{L}(ORG|s)$ where s is two-character long and the curve of weighted f-measure $F_{0.5}$. (b) The distribution of $\mathcal{L}(ORG|s)$ where s is three-character long and the curve of weighted f-measure $F_{0.33}$.

The best f-measure and the corresponding excluding rate are about 0.88 and 0.90

with two-character-long organization names, and about 0.96 and 0.98 with three-character-long ones. It is very obvious that the excluding rate and recall rate of organization names is much lower than other models. A heuristic strategy is adopted to alleviate the false detecting problem. If the first or the last character of some three-character-long candidate is a monosyllabic word that often appears adjacent to a name, as “前遠東” and “東鼎與”, this candidate will be removed from the candidate pool.

4.6. Location-like NE Model

The location-like named entity model is used to recognize whole location-like named entities. As we mentioned before, location-like NEs denote NEs with similar corresponding *genuine name* types to location suffixes. These include all types of location names and racial names. The corresponding *genuine name* types comprise location and transliteration name candidates. Therefore, basically we use a simple rule to recognize whole location-like NEs:

Location-like NE = (Location or Transliteration Name Candidate) + Suffix

Because suffixes could provide confidences, if the likelihood of location name candidates is over half of the original threshold of $\mathcal{L}(LOC|s)$, the rule will be applied. (Transliteration name candidates do not practice this rule.) For example, $\mathcal{L}(LOC|$*“武夷”*$)$ is only 0.217, which is lower than the threshold of location name, 0.24. However if “武夷” is followed by the suffix “山”, since $\mathcal{L}(LOC|$*“武夷”*$)$ is more than 24/2=12, “武夷山” will be added into the candidate list of whole location name accordingly.

Some often seen location-like suffixes are listed in Appendix C.

4.7. Organization-like NE Model

The organization-like named entity model is responsible for the recognition of whole organization-like named entities. As we mentioned before, location-like NEs denote NEs with similar corresponding *genuine name* types to organization suffixes. These include all organization names except for racial names, facility names, publication names, part of miscellaneous names, etc. The corresponding *genuine name* types comprise organization, transliteration, and personal name candidates.

The situation with organization-like NEs is more complicate than that with location-like NEs. The internal structure of a whole organization named entity is much more complex than other NEs [19]. Besides the simplest suffixes denoting the type of NEs like “公司” and “協會”, sometimes there are other suffixes or prefixes describing the properties of NEs, such as “蘋果 **電腦** 公司” and “**財團法人** 董氏基金會”. Usually these descriptive affixes are also common words related to the type of NEs it describes; moreover, *genuine names* of organizations are more similar to common words than other named entities; hence the confidence that could be brought by descriptive affixes is limited. Concerning “國際鋼鐵市場” and “台灣電腦公司”, the appearance of descriptive affixes instead increases the chance of being wrongly recognized.

Since descriptive prefixes are rarely used in documents unless very formal situations and they could be treated in a similar way to suffixes, we only focus on descriptive suffixes in this paper. Generally speaking, descriptive affixes can be divided into two types. The denotations of company names are borrowed here: one type is called *business description* and the other is called *affair description*. The distinction between them is that the former denotes a more concrete and specific item, and the latter represents a more generic category. For example, in “台灣塑膠工業股

份有限公司”, “塑膠” is a business description and “工業” is an affair description; in “台北市旅行商業同業公會”, “旅行” is a business description and “商業” is an affair description. Generally business descriptions are put in front of affair descriptions; thus we mainly use the following rule to recognize organization-like NEs:

Organization-like NE = (Organization Name or Transliteration Name or
Personal Name Candidate) + [Business description]$^{+}$
+ [Affair description]$^{+}$ *+ Typing suffix*

Some often seen organization-like suffixes are listed in Appendix C.

The confidence that could be brought by the three kinds of suffixes is different. Here we consider the confidence as a “magnification”, i.e. suffixes with a confidence *n* could bring additional *n*-times of $\mathcal{L}(LOC|GenuineName)$. For example, if the confidence of the suffixes is three, it only needs quarter of the original likelihood to go beyond the threshold that allows a *genuine name* to be attached with these suffixes to form a whole NE candidate. In other words, the confidence of location-like suffixes mentioned in previous section can be view as one. The threshold of whole organization-like NE candidates is set as same as that of *genuine names* (0.2/0.1).

Because of the problems mentioned before, the confidence of three types of suffixes is set as follows. Business descriptions cannot bring any confidences; affair descriptions longer than one character have a confidence of one; typing suffixes constituted of two characters or monosyllabic ones that appear together with one of other two types of suffixes have a confidence of one. Typing suffix longer than two characters could brings twice of confidences. For instance, the confidence of “錢櫃傳播事務有限公司” is:

1 *(genuine name) +* ***0*** *(business description) +* ***1*** *(affair description)*

+ ***2*** *(typing suffix) =* ***4***

That is to say, as long as $\mathcal{L}$*(LOC| “錢櫃”)* is more than 0.2/4 =0.05, “錢櫃傳播事務

有限公司” will be added into the candidate list.

4.8. Abbreviation Model

The abbreviation model is used to recognize abbreviations and some rule-recognizable aliases. A simple rule-driven approach is applied to complete this job. Except for non-suffixed transliteration names, all other kinds of candidates extracted by above models are checked by heuristic rules to see if there are strings in the document fitting in these generating rules. If someone is found, it will be added into the candidate list. These rules are listed below:

Rule 1: Take the first characters of genuine name and suffixes other than typing suffix, and the last character of typing suffix from whole NE candidates (e.g. “中央 研究 院” → “中研院”)

Rule 2: Surnames of personal name candidates (e.g. “呂秀蓮” → “呂”)

Rule 3: Given names of personal names (e.g. “陳信安” → “信安”)

Rule 4: *Modifier* + *Surname* or *any character of Given names* (e.g. “陳水扁” → “小陳”, “阿水”, “阿扁”, etc.)

Notice that only single entity abbreviations and aliases with original references appearing in the document could be addressed by the model. The recognition of combined abbreviations and abbreviations without original references is a tough problem, and it is beyond our scope. The third type of abbreviations, derived named entities, is handled by whole named entity model as much as possible. For example, “市長” is added into location-like suffixes, and “部長” is added into organization-like suffixes. Thus if the *genuine name* could be correctly recognized, most of the merging type of derived NEs can be solved. Consider the case “台北市長”:

台北(location name) + 市長(abbreviation suffix) → 台北市長(derived NE)

Chapter 5 Lexical Analysis & Recovery

The lexical analyzer is responsible for verifying candidates generated by the candidate generator. Heuristic rule-driven approaches are adopted to filter out false named entity candidates and resolve ambiguities caused by false candidates. There are six heuristic rules applied in order precedence:

Heuristic Rule 1: Tri-word maximal matching (To be discussed in Section 5.1)

Heuristic Rule 2: Least number of NEs first (To be discussed in Section 5.2)

Heuristic Rule 3: Most frequently appearing NEs first (To be discussed in Section 5.3)

Heuristic Rule 4: Words of even lengths first (To be discussed in Section 5.4)

Heuristic Rule 5: Often seen monosyllabic words first (To be discussed in Section 5.5)

Heuristic Rule 6: Forward precedence (To be discussed in Section 5.6)

The recovery mechanism is used to revive some obvious incorrect results of the lexical analyzer, which is not suitable to be solved by priority-style rules that the lexical analyzer adopts. A rule-driven segmentation checker is developed to recover correct named entities that are wrongly filtered by candidate generator or lexical analyzer and to remove false candidates that are not discovered by lexical analyzer. The recovery mechanism is discussed in Section 5.7. Section 5.8 gives some experiments with above approaches to examine their performance.

5.1. Tri-word Maximal Matching

The tri-word-maximal-matching rule is proposed by Chen & Liu (1992) [6]. The rule follows below three steps:

1. From the segmenting point in the document, look forward for all possible three-word combinations.
2. Take the first word of the longest sequence of all, segmenting this word.
3. Move to next segmenting point.

For example, with the sentence “張大春天天看報紙”, the tri-word maximal matching may work like this:

1.	張 大 春 天天看報紙 張 大 春天 天看報紙 張大 春 天 天看報紙 張大 春 天天 看報紙 張大 春天 天 看報紙 張大春 天天 看 報紙	Choose “張大春” as the segmentation result, because “張大春 天天 看 報紙” has the longest tri-word sequence.
2.	張大春 天 天 看 報紙 張大春 天天 看 報 紙 張大春 天天 看 報紙	Choose “天天” as the segmentation result, because “天天 看 報紙” has the longest tri-word sequence.
3.	張大春 天天 看 報 紙 張大春 天天 看 報紙	Choose “看” as the segmentation result, because “看 報紙” has the longest tri-word sequence.
4.	張大春 天天 看 報 紙 張大春 天天 看 報紙	Choose “報紙” as the segmentation result, because “報紙” has the longest tri-word sequence.
5.	張大春 天天 看 報紙	Obtain the whole segmentation results.

The tri-word maximal matching can generally solve over 95% of the segmentation ambiguities in our experimental results, which will be discussed in Section 5.8. However, there are still some situations that tri-word maximal matching cannot help. Sometimes there are many tri-word segmentation sequences with same length. For example, “查理王光輝的一生”, the two ambiguous tri-word segmentation sequences “查理王 光輝 的 一生” and “查理 王光輝 的 一生” have the same length and cannot be solved by using maximal-matching rule. In these situations, the second rule, least number of NEs first, would be applied.

5.2. Least Number of NEs First

In the cases which cannot be solved by the first rule, the one with the least number of named entities will be chosen among all tri-word sequences of the same length. For above example, “查理王 光輝 的 一生” has one NE candidate and “查理 王 光輝 的 一生” has two, so the system would choose the former one as the right answer.

This rule can effectively filter out about 64% of false candidates which cannot be solved by the first rule and chances of incorrect filtering are less than 1%. However, sometimes ambiguous tri-word sequences of the same length might have same number of NEs. For example, “張宇 成功 的 時候” and “張宇成 功 的 時候” have same length and same number of named entities. Thus the maximal matching rule and the least NEs rule cannot solve the ambiguity of these two sequences.

5.3. Most Frequently Appearing NEs First

Sometimes people rely on re-appearances of names to understand vague sentences. The most-frequently-appearing-NEs-first rule follows a similar way and picks the tri-word sequence with the most appearing times of component NEs in the input document. For above example, if “張宇” appears elsewhere and is not followed by the character “成” in the document, “張宇 成功 的 時候” will be picked as correct segmentation.

This rule can solve not only covering ambiguities like above, but also overlapping ambiguities. Consider the example “台大連醫學院在內”, “台大” and “大連” are both recognized as location names. If the input document is an article about “台大”, generally the appearance times of “台大” would be much more than that of “大連”. Therefore, “台大 連 醫學院 在內” will be picked as correct segmentation according to the most-frequently-appearing-NEs-first rule.

5.4. Words of Even Lengths First

The word-of-even-lengths-first rule is modified from the average-length rule proposed by [6]. Modern Chinese generally prefer to use words of even lengths. In our lexicon, nearly 70% of entries have even lengths. This gives us some clues to resolve the segmentation ambiguity. The average-length rule exploits the even-length preference and assumes that the tri-word sequence with a minimal variance is supposed to be the right segmentation. However, a sequence of three words all composed of three characters is actually stranger than a 3, 2, 4 sequence.

In our system, we modify the average-length rule to choose the sequence with most words of even lengths. For the above example, “張宇成功的時候”, if “張宇” is not appears elsewhere and the most-frequently-appearing-NEs-first rule does not work, “張宇 成功 的 時候” will be chosen by the even-length rule because it has more words of even lengths.

There are several exceptional conditions of this rule. First, personal names, transliteration names, and numerical expressions are not concerned in this rule. In another words, “張宇” in the above example is not considered as a word of even lengths. Second, the often seen monosyllabic words, like “的”, “之”, “也”, etc., are viewed as words of even lengths instead. That is to say, “張宇 成功 的 時候” is regarded as totally having two words of even lengths, one is “成功” and another one is “的”, not “張宇”. Third, the suffix part of a suffixed named entity is not considered into the length of the whole named entity. For example, “揚昇高爾夫球場” is viewed as a word of even lengths, not of odd ones.

Consider another example, “黃大目 的 豆干” and “黃大 目的 豆干”, any of the four rules we have mentioned cannot solve this ambiguity. To address this situation, the fifth rule is introduced in this paper.

5.5. Often Seen Monosyllabic Word First

The often-seen-monosyllabic-word-first rule is also proposed by [6]. When the above four rules cannot solve the ambiguity, the sequence with the most often seen monosyllabic words is picked. For the above example, “黃大目 的 豆干” has one often seen monosyllabic word and “黃大 目的 豆干” has none, hence the former will be chosen by the often-seen-monosyllabic-word-first rule.

The problem is which words are qualified to be often seen. Chen & Liu’s collection adding some other monosyllabic words is adopted in our system. These words are listed in Appendix D.

5.6. Forward Precedence

When all heuristic rules discussed above cannot resolve the ambiguities, the forward precedence is applied, i.e. the tri-word sequence with longer forward words would be picked. For example, with two ambiguous tri-word sequence “決戰 爭 勝負” and “決 戰爭 勝負”, the former would be picked since “決戰” is longer than “決”.

In our experimental results, while all above five heuristic rules cannot make choices, the accuracy of left precedence is nearly 80%. For this reason and the easy implementation with forward maximal-matching, the left precedence is adopted in our system instead of right precedence.

5.7. Recovery Mechanism: Segmentation Checker

Inevitably, there are over-generations and under-generations in the outputs of the first two phases of our system. Sometimes these mistakes would cause lexical anomalies in the segmentation result. These anomalies mainly comprise two situations. The first one, over-segmentations caused by under-generations, is due to true named entities not recognized by the candidate generator. The second one, on the contrary, under-segmentations caused by over-generations, results from false candidates not filtered out by the lexical analyzer. The segmentation checker would detect these anomalies to find suspect segmentation sequences and try to recover the incorrect recognition of NEs.

To deal with the over-segmentation case, all sequences constituted of three or more seldom used monosyllabic words in a row are suspected. These suspect sequences will be checked to see if any fragments of them could constitute named entity candidates with $\angle(TYP|s)$ over a predefined threshold of the corresponding type. (To distinguish from the threshold mentioned in Chapter 3, we will use “suspect threshold” and “candidate threshold” to refer these two different thresholds respectively in following paragraphs.)

For example, since $\angle(TRA|\text{“龐畢度”}) = 0.43 < 0.51$, the candidate threshold of $\angle(TRA|s)$, the string is usually segmented to “龐 畢 度” in the first two phases. This suspect sequence will be detected by the segmentation checker. Because $\angle(TRA|\text{“龐畢度”})$ is larger than the suspect threshold of $\angle(TRA|s)$, which is set to 0.2 in our system, “龐畢度” is added into the candidate list of transliteration names.

With personal names, there is another special case of over-segmentations. Let us consider the personal name “陳水扁”. $\angle(PER|\text{“陳水扁”}) = 0.23 < 0.26$, the candidate threshold of $\angle(PER|s)$. However, $\angle(PER|\text{“陳水”})$, which equals 0.55, is larger than the candidate threshold. When this situation happens, the personal name is usually incorrectly segmented into a personal name of two characters and a monosyllabic word, such as “陳水 扁” in this case. To cope with this situation, the following sequence is also viewed as suspects of over-segmentations:

two-character-long personal name candidate + seldom used monosyllabic word

On the other hand, to deal with under-segmentations, all segmentation sequences constituted of interlaced appearances of transliteration names, location names, organization names, and seldom used monosyllabic words, are suspected. These suspect sequences are attempted to be re-segmented into a new sequence containing one more word than the original segmentation sequences. For example, if “群中” is incorrectly recognized as a location name, the phrase “台北人群中” would be wrongly segmented into a suspect sequence “台北人 群中”. The suspect sequence would be detected by segmentation checker and be re-segmented into the right sequence “台北 人群 中”. If the re-segmenting cannot be performed, the original sequence will be kept.

The procedure of segmentation checker is as follows:

1. Check over-segmented sequences
2. Check under-segmented sequences
3. Repeat step 2, until no new suspect sequences appear
4. Check over-segmented sequences again

5.8. Discussions about Heuristic Rules

The test samples of our system (61 news articles from United Daily News and Central News Agency, which will be further discussed in Chapter 6) are used to measure the performance of our heuristic-rule-driven lexical analyzer on ambiguity resolution. The following measurements are adopted:

- Ambiguous Tri-Word Sequences: # of all possible tri-word sequences which could not be discriminated by the prior rules
- Resolved: # of tri-word sequences which could be filtered by the corresponding heuristic rule
- Errors: # of correct words which are wrongly filtered
- Applying Rate: Resolved / Ambiguous Tri-Word Sequences
- Accuracy: 1 – Errors / Resolved

The experimental results are listed in Table 5.1:

Table 5.1 The performance of heuristic rules in ambiguity resolution.

	Ambiguous Tri-Word Sequences	Resolved	Errors	Applying Rate	Accuracy
Heuristic Rule 1	81273	78263	265	96.30%	99.66%
Heuristic Rule 2	3010	1935	10	64.29%	99.48%
Heuristic Rule 3	1075	225	5	20.93%	97.78%
Heuristic Rule 4	850	603	20	70.94%	96.68%
Heuristic Rule 5	247	9	1	3.64%	88.89%
Heuristic Rule 6	238	238	49	100.00%	79.41%

Comparing with the same experiment mentioned in [6], we have a similar applying rate and accuracy to [6]’s experiment with maximal-matching rule. Our rest rules outperform theirs whether in applying rate or accuracy. However, two experiments take different assumptions that [6] assume that there are no OOV words but the second and the third rule of ours are based on the recognition of OOV words. Therefore, the results are less comparable.

Chapter 6 Evaluation

Our system is developed in Microsoft Visual Basic .NET, and measured on a Pentium 4 2GHz CPU with 512 MB ram. Our test samples are 61 articles of different topics acquired from United Daily News and Central News Agency.

6.1. Experiments and Results

To measure the performance of our system, a corpus which is balanced and well-tagged according to our standard is needed. However, there is no such tagged corpus conforming to our needs available. Therefore, instead of a standard testing corpus, we obtain 61 articles from United Daily News and Central News Agency as our test bed. These articles are segmented and tagged by our system, and corrected manually.

These 61 articles are gathered from five different domains in which NE types that our paper focus on are important and often seen. These five domains are politics, society, business, sports, and entertainment. Because the quantity of politics news and society news is more than other domains, we obtain three different sub-topics (lawsuit, government, and election) from politics news and two (crime and local) from society news.

Table 6.1 draws the total number of true NEs in test documents, NE candidates extracted, and false candidates in extracted ones. Standard measurements of *recall* and *precision* are estimated:

$$\mathit{Recall} = (\#\ \mathit{of\ Ext.} - \#\ \mathit{of\ False})/(\#\ \mathit{of\ True})$$

$$\mathit{Precision} = 1 - (\#\ \mathit{of\ False})/(\#\ \mathit{of\ Ext.})$$

Notice that there are two special columns in the table, number of words and excluding rate. Because appearing frequencies of NEs in different domains are varied and actually have a great impact on precision, precision is thus less meaningful. We think that excluding rate might be a better measurement of over-generation. Excluding rate is counted from:

Excluding = 1 - (# of False)/(# of Words - # of True)

It stands for the percentage of non-NEs being correctly filtered by our system.

Table 6.1 Experimental results of our system.

Topic	Articles	True	Extracted	False	Words	Recall	Precision	Excluding
Politics 1	7	211	224	34	3460	90.05%	84.82%	98.95%
Politics 2	10	465	444	32	4343	88.60%	92.79%	99.17%
Politics 3	7	158	155	18	2750	86.71%	88.39%	99.31%
Society 1	7	321	317	23	3599	91.59%	92.74%	99.30%
Society 2	10	372	378	39	5423	91.13%	89.68%	99.23%
Business	7	295	289	34	3392	86.44%	88.24%	98.90%
Sports	7	272	226	18	3690	76.47%	92.04%	99.47%
Entertainmen	6	196	182	16	2742	84.69%	91.21%	99.37%
Total	61	2290	2215	214	29399	87.38%	90.34%	99.21%

Table 6.2 shows the recall rates of different types of NE in different domains. Because our system does not focus on automatic classification of NEs, one NE might be recognized by different models. For example, “戴高樂” is recognized by both personal name model and transliteration model. Besides, there are vague areas among different type of NEs. For instance, “新新聞” is usually a publication name, however in the phrase “新新聞 社長”, “新新聞” could be also viewed as an organization name. Therefore, it’s hard to judge the precision of each type and only the recall rates are listed here.

Table 6.2 Recall of our system with different types of NEs.

Topic		PER	TRA	LOC	ORG	ABB	PO	LO	OO	AO	TITLE	MIS
Politics 1	True	104	2	14	9	6	0	0	2	1	66	5
	Detected	102	2	9	5	6	0	0	2	0	61	2
	Recall	98.08%	100.00%	64.29%	55.56%	100.00%	--	--	100.00%	0.00%	92.42%	40.00%
Politics 2	True	261	0	18	11	9	5	0	148	10	4	0
	Detected	250	0	17	5	6	5	0	128	2	2	0
	Recall	95.79%	0.00%	94.44%	45.45%	66.67%	100.00%	--	86.49%	20.00%	50.00%	--
Politics 3	True	43	0	24	0	2	0	0	42	46	2	0
	Detected	43	0	24	0	2	0	0	28	43	0	0
	Recall	100.00%	0.00%	100.00%	0.00%	100.00%	--	--	66.67%	93.48%	0.00%	--
Society 1	True	185	1	115	0	30	2	1	2	0	1	0
	Detected	180	1	98	0	29	1	1	2	0	0	0
	Recall	97.30%	100.00%	85.22%	0.00%	96.67%	50.00%	100.00%	100.00%	--	0.00%	--
Society 2	True	135	3	137	13	6	2	8	60	5	0	2
	Detected	133	2	128	11	6	2	7	43	3	0	2
	Recall	98.52%	66.67%	93.43%	84.62%	100.00%	100.00%	87.50%	71.67%	60.00%	--	100.00%
Business	True	72	2	65	131	0	0	9	4	5	1	0
	Detected	68	2	56	111	0	0	3	4	5	0	0
	Recall	94.44%	100.00%	86.15%	84.73%	0.00%	--	33.33%	100.00%	100.00%	0.00%	--
Sports	True	56	110	23	9	1	4	6	58	3	1	5
	Detected	50	99	19	8	1	0	6	20	2	1	4
	Recall	89.29%	90.00%	82.61%	88.89%	100.00%	0.00%	100.00%	34.48%	66.67%	100.00%	80.00%
Entertainment	True	126	12	18	4	4	12	1	0	5	12	4
	Detected	118	9	16	2	4	6	1	0	5	7	1
	Recall	93.65%	75.00%	88.89%	50.00%	100.00%	50.00%	100.00%	--	100.00%	58.33%	25.00%
Total	True	982	130	414	177	58	25	25	316	75	87	16
	Detected	944	115	367	142	54	14	18	227	60	71	9
	Recall	96.13%	88.46%	88.65%	80.23%	93.10%	56.00%	72.00%	71.84%	80.00%	81.61%	56.25%

Notice that the first five columns (PER, TRA, LOC, ORG, ABB) only include the focused types of our system. Column PER comprise only formal Chinese personal names and personal names with appellations. Other personal names, such as Japanese name “酒井光次郎” and pseudonym “老子”, are counted in column PO instead. Monosyllabic place names without suffixes, like “粵” and “台”, are recognized by lexicon matching and counted in column LO. Government and team names are also recognized by lexicon. They are viewed as OO. All other location names and organization names are included in column LOC and ORG respectively. Column ABB contains only abbreviations with original reference in the input document, other abbreviations are considered as AO.

In table 6.2 we can see that our system obtain a high recall of 96.13% with formal personal names, and satisfied recall rates with other types. Next section will further discuss meanings of our experimental results.

6.2 Discussions

Misrecognitions of our system could be categorized into six reasons, which will be detailed individually in following paragraphs. The distributions of these errors are listed in Table 6.3. You may refer several examples of erroneous sentences in our testing data with their reasons of misrecognition in Appendix E.

Table 6.3 The error analysis of misrecognitions.

Topic	Statistical Deviations		Misleading Suffixes	OOV	Insufficiencies of		Shortcomings of Heuristic	Inherent Ambiguities	Sum
	Over-Est.	Under-Est.			Non-Focus	Suffixes & Structures			
Politics 1	8	3	9	1	9	8	9	0	47
Politics 2	13	4	11	4	30	8	10	0	80
Politics 3	6	0	6	1	18	0	4	0	35
Society 1	10	2	6	0	2	18	11	2	51
Society 2	16	5	27	1	20	16	9	0	94
Business	8	2	13	3	7	8	15	0	56
Sports	11	4	10	14	38	2	23	2	104
Entertainment	11	7	5	3	14	5	4	1	50
Total	83	27	87	27	138	65	85	5	517
Percentage	16.05%	5.22%	16.83%	5.22%	26.69%	12.57%	16.44%	0.97%	100.00%

■ Statistical Deviations

Statistical deviations denote errors caused by simplified stochastic models and imprecise estimations of statistics. They can be further divided into two types: over-estimating and under-estimating. There are about 16% of misrecognitions in our test sample caused by over estimating deviations, and about 5% caused by under-estimating.

The over-estimating situation is due to the over-simplified unigram model we adopt. For example, “爾”, “文”, “雅” 溫” are all like transliterating characters. “雅爾文” and “雅爾溫” are actually like transliteration names. But how about “溫文爾雅”? If bi-gram models are used, the number of this type of false positive errors would much decrease.

Under-estimating situations happen because our training data are not large enough. There is inevitably some deviation with rarely naming characters. Although the recovering mechanism could help partially, there are still some cases cannot be revived. For example, personal name “石金受” is wrongly recognized as “石金 受” because “受” does not appear in our training data of personal names. Unfortunately, “受” is a frequently used single-character word, too. Therefore, the recovering mechanism cannot revive it. Another example is transliteration name “簡屈”. It is wrongly segmented as “簡 屈” in our system due to the low *like(c, TRA)* of “簡” and “屈”. Our recovering mechanism only checks segmentation sequence constituted of successive three or more single-character words and “簡 屈” only has two, so “簡屈” cannot be revived.

■ Misleading Suffixes

Suffixes do not always bring confidences but might be sometimes misleading. That is because suffixes of NE generally can be also used as common words. If the previous word is like a name, these words might be wrongly attached to the previous word as a whole NE candidate and cannot be filtered or recovered. For example, “國際市場” is wrongly recognized as a name of some market. In fact, we cannot solve this type of ambiguity without contextual information or syntactical and semantic analysis. However, there are about 17% of errors are caused by misleading suffixes.

The most serious case is that with frequently used monosyllabic words “教” and “會”. When these two words are used as common verbs, their subjects are often people. However, people’s name might be also suffixed with these two words to form names of some religion or association. For example, “郭富城 教 大家” is always wrongly recognized as “郭富城教 大家” in our system.

■ Out-of-Vocabulary Words

Intuitively, out-of-vocabulary words often cause errors in segmentation because the system does not know that they are words. These errors might be propagated since most segmentation methods rely on compromises with context to perform ambiguity resolution. Therefore, while there are OOV words adjacent to false candidates, our system might not perform filtering correctly. There are about 5% of errors are caused by OOV words.

We know that it is impossible to compile all words in a lexicon. OOV words always exist in documents. Besides NEs which our system would recognize, there are still un-recognized NEs, derived words, and other OOV words. Usually an un-recognized OOV word would be segmented into sequences of single characters. If some characters of an OOV word are wrongly recognized as part of adjacent false NE candidates, our system could not filter these false candidates anyway. Sometimes our system would recognize a non-NE OOV words as a NE. Since this situation brings few adverse effects, we will neither count it as a true positive result nor a false one in our evaluation.

■ Insufficiencies of Human Knowledge

Insufficiencies of human knowledge constitute the major part of errors of our system. There are two main types of insufficiencies of handcrafted knowledge. The first one is the NE types we do not focus on, such as abbreviations without original references and titles without suffixes. This type of insufficiencies account for about 27% of errors of our system.

The second type of insufficiencies is the set of typing suffixes and the over-simplified structures of whole location names and organization names. It account for about 13% of errors of our system. Because there are too many types of organizations and we don't have a vast training data of each type of them, many true negative errors with organization names are due to that our system do not realize some word could be a suffix. For example, "台灣野鳥協會", since "野鳥" is not collected in the set of business suffixes, our system could not recognize "台灣野鳥協會" as an organization name.

On the other hand, our system adopts over-simplified structures to address whole named entities, there are many cases not being considered. For example, "國立交通大學" is segmented into "國立 交通大學" by our system because the appearance of prefix "國立" is not in our considerations. Lei (2003) [19] have proposed a thorough analysis of organization names, and it can be used in our future improvement.

■ Shortcomings of Heuristic Rules

All "heuristic" rules have their shortcomings. As we have discussed in the previous chapter, heuristic rules might sometimes choose wrong segmenting sequences or incorrectly suspect some innocent ones. Errors caused in these situations account for about 16% of total errors.

The covering ambiguity problem is a typical kind of shortcomings of heuristic rules. As experiments show that, maximal-matching-rule-driven method could effectively resolve overlapping ambiguities, but not covering ones. If a false candidate is constituted of two or more true words, our lexical analyzer could not filter it out. For example, "喝 咖啡 的 錢 都 能" is wrongly segmented into "喝 咖啡 的 錢都 能" and cannot be recovered. Sometimes this kind of false candidates degrades recall, too. For instance "連 友達 都 決定" is incorrectly recognized as "連友達 都

決定”. In this situation, we do not only have a false candidate, but also lose a true NE.

Another serious problem caused by heuristic rules is that if some named entity is also a common word, this NE will be wrongly discarded by the second rule of lexical analyzer. “連戰” is a typical example of this kind of NEs.

■ Inherent Ambiguities

Some segmentation ambiguities are “inherent”. That is to say, these ambiguities are caused by ambiguous sentences, not faults of our system. For example, “看安東尼大展身手”, two interpretations, “看 安東尼 大展身手” and “看 安東尼大 展身手”, are all correct whether in syntactic or semantic. The most frequently appearing rule and the team algorithm of transliteration names are designed to tackle this situation. However, if the true candidates “安東尼” does not appear elsewhere in the input document, these two mechanism are totally helpless. There are about 1% of errors of our system caused by inherent ambiguities.

Chapter 7 Conclusions and Future Works

Our system focuses on the recognition of named entities to solve out-of-vocabulary problem in natural language processing. In this thesis, we proposed a multi-layered “generation, filtering, and recovery” framework to tackle this problem. In the generation phase, four types of genuine names (PER, TRA, LOC, ORG), two types of whole NEs (LOC-like, ORG-like), and abbreviations are extracted by statistical models and rule-driven approaches. We tune the models to obtain a high recall rate in the first phase. In the second phase, a maximal-matching-rule-driven lexical analyzer is adopted to perform ambiguity resolution, which shows a great power in the false candidate filtering. Simple heuristic rules are applied in the third phase, which can effectively detect obvious over-segmentations and under-segmentations in the results of the first two phases.

Overall speaking, pure lexical information is employed to recognize named entities in our system. Only statistical features and internal structures of NE are utilized. Our statistical model and heuristic rules are simplified for easy implementation. Although our system gets a satisfied performance, there are still many rooms for improvement. First, statistical models could be refined. More training data could be collected. More elaborate candidate generating model could be adopted, such as bi-gram model. More internal features could be exploited, such as positional information of characters. Contextual information, such as word probability of being adjacent to some type of NEs, could be added into our model, too.

Second, heuristic rules could be more completed or substituted by other mechanism. As we mentioned in previous section, shortcomings of heuristic rules form an upper-bound barrier of performance. We could introduce more heuristic rules to recover the inadequacy of original ones, or use other mechanism like statistical approaches to replace rule-driven methods.

Third, more candidate generating models could be added. Many types of NEs have not been addressed in our system. In the experiment results we could find that these NEs occupy a great proportion of true negative errors. If these NEs could be recognized, the recall rate of our system is supposed to be boosted.

Fourth, more knowledge could be gathered and utilized. The suffix and appellation information used in our system is handcrafted at present. Bootstrapping algorithm might help us automatically retrieve these kinds of information from the Internet. Part-of-speech tagging, syntactic checking and even semantic analysis might be added into our future system, too.

The multi-layered "generation, filtering, and recovery" framework we proposed let us easily substitute the design and implementation of individual layers and incorporate above refinements to achieve the ultimate performance under this framework.

References

[1] Bloomfield, Leonard, 1933, “Language,” Chicago: University of Chicago Press.

[2] Chang, Jyun-Sheng, C. D. Chen, and S. D. Chen, 1991, “Chinese Word Segmentation through Constraint Satisfaction and Statistical Optimization,” (in Chinese) Proceedings 23 of ROCLING-IV, ROC Computational Linguistics Conferences, pp. 147-165, Kenting, Taiwan, ROC.

[3] Chang, Jyun-Sheng, S. D. Chen, Y. Zheng, X. Z. Liu, and S. J. Ke, 1992, “Large-Corpus-Based Methods for Chinese Personal Name Recognition,” Journal of Chinese Information Processing, 6(3), pp. 7-15.

[4] Chen, Hsin-Hsi and J. C. Lee, 1994, “The Identification of Organization Names in Chinese Texts,” Communication of COLIPS, Vol.4 No. 2, 131-142.

[5] Chen, Hsin-Hsi and J. C. Lee, 1996, “Identification and Classification of Proper Nouns in Chinese Texts”, Proceedings of Coling-96, Vol. 1, pp. 222-229.

[6] Chen, Keh-Jiann and S. H. Liu, 1992, “Word Identification for Mandarin Chinese Sentences,” Proceedings of COLING-92, Vol. 1, pp. 101-107

[7] Chen, Keh-Jiann, 1999, “Lexical Analysis for Chinese- Difficulties and Possible Solutions,” Journal of Chinese Institute of Engineers, Vol. 22. #5, pp. 561-571

[8] Chen, Keh-Jiann and C. J. Chen, 2000, “Knowledge Extraction for Identification of Chinese Organization Names,” Proceedings of ACL workshop on Chinese Language Processing, pp.15-21.

[9] Chiang, Tung-Hui., J. S. Chang, M. Y. Liu, & K. Y Su, 1996, “Statistical Word Segmentation,” in C. R. Huang, K. J. Chen and B. K. Tsou (Eds.), Readings in Chinese natural language processing, pp. 123-146, Journal of Chinese Monograph Series Number 9.

[10] Chien, Lee-Feng, 1999, “PAT-Tree-Based Adaptive Keyphrase Extraction for Intelligent Chinese Information Retrieval,” Inf. Process. Manage. 35(4), pp. 501-521

[11] Chinchor, Nancy and P. Robinson, 1998, "MUC-7 Named Entity Task Definition (version 3.5)," in Proceedings of the MUC-7.

[12] CKIP, 1995, "Sinica Corpus," technical report 95-02, Academia Sinica, Taipei.

[13] Chua, Tat-Seng and J. Liu, 2002, "Learning Pattern Rules for Chinese Named Entity Extraction," Proceedings of AAAI/IAAI 2002, pp. 411-418

[14] Dong, Zhen-Dong and Q. Dong, 2000, "HowNet," http://www.keenage.com/zhiwang/e_zhiwang.html.

[15] GB/T 13715-92, 1993, "Contemporary Chinese Language Word-Segmentation for Information Processing," technical report, Beijing.

[16] Goh, Chooi Ling, M. Asahara, Y. Matsumoto, 2003, "Chinese Unknown Word Identification Using Character-based Tagging and Chunking," ACL-2003 Interractive Posters/Demo, pp. 197-200

[17] Huang, Chu-Ren, K. J. Chen, and L. L. Chang, 1997, "Segmentation Standard for Chinese Natural Language Processing," Computational Linguistics and Chinese Language Processing, 2.2, pp: 47-62.

[18] Ji, Heng and Z. S. Luo, 2001, "Inverse Name Frequency Model and Rule Based Chinese Name Identification," (In Chinese) Natural Language Understanding and Machine Translation, Tsinghua University Press, pp. 123-128.

[19] Lei Jing, 2003

[20] Loukachevitch, Natalia and B. Dobrov, "Sociopolitical Domain As a Bridge from General Words to Terms of Specific Domains," Proceedings of GWC 2004, pp. 163-168

[21] Luo, Zhi-Yong and R. Song, "Integrated and Fast Recognition of Proper Noun in Modern Chinese Word Segmentation," Proceedings of International Conference on Chinese Computing, 2001, Singapore, pp. 323-328.

[22] Mo, Ruo Ping, Y. J. Yang, K. J. Chen, and C. R. Huang, 1996,

"Determinative-Measure Compounds in Mandarin Chinese Formation Rules and Parser Implementation," In C. R. Huang, K. J. Chen and B. K. Tsou (Eds.), Readings in Chinese natural language processing, pp. 123-146, Journal of Chinese Monograph Series Number 9.

[23] Nguyen, Trien T., 2000, "VECONREF: Building an Online Database of Research in Vietnamese Economics," http://www.arts.uwaterloo.ca/~vecon/veconref/vecondoc.pdf (documentation)

[24] Packard, Jerome, 2000, "The Morphology of Chinese: A Linguistic and Cognitive Approach," Cambridge University Press, Cambridge.

[25] Sekine, Satoshi, K. Sudo, and C. Nobata, 2002, "Extended named entity hierarchy," Proceedings of the LREC 2002 Conference, pp. 1818-1824.

[26] Shen, Da-Yang and M. S. Sun, 1995, "Chinese Location Name Recognition," (In Chinese) Development and Applications of Computational Linguistics. Tsinghua University Press.

[27] Song, R, H. Zhu, W. Pan, and Z. Yin, 1993, "Automatic Recognition of Person Names Based on Corpus and Rule Base," Research and Application of Computational Linguistics, Beijing Language Institute Press, Beijing, China.

[28] Sproat, Richard, C. Shih, W. Gale, and N. Chang, 1996, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese," Computational Linguistics, 22.3, pp.377-404.

[29] Sun, Jian, J. F. Gao, L. Zhang, M. Zhou, and C. N. Huang, 2002, "Chinese Named Entity Identification Using Class-based Language Model," Proceedings of the 19th International Conference on Computational Linguistics, Taipei, pp. 967-973

[30] Sun, Mao-Song, 1993, "English Transliteration Automatic Recognition," In Computational Language Research and Development, L. W. Chen & Q. Yuan, ed., Beijing Institute of Linguistic Press.

[31] Sun, Mao-Song, C. N. Huang, H. Y. Gao, and J. Fang, 1994, "Identifying Chinese Names in Unrestricted Texts," Communication of COLIPS, Vol.4 No. 2,

pp. 113-122

[32] Sun Mao-Song, D. Y. Shen, and C. N. Huang, 1997, “CSeg&Tag1.0: A Practical Word Segmenter and POS Tagger for Chinese Texts,” Proceedings of the 5th Int’l Conference on Applied Natural Language Processing, Washington DC, USA.

[33] Tan. Hong-Ye, 1999, “Chinese Place Automatic Recognition Research,” Proceedings of Computational Language, C. N. Huang & Z.D. Dong, ed., Tsinghua Univ. Press, Beijing, China.

[34] Van Rijsbergen, Keith, 1979, “Information Retrieval,” Butterworth.

[35] Wu, Andi and Z. Jiang, 1998, “Word Segmentation in Sentence Analysis,” Proceedings of the 1998 International Conference on Chinese Information Processing, Beijing, China.

[36] Wu, Andi, 2003, “Customizable Segmentation of Morphologically Derived Words in Chinese,” Computational Linguistics and Chinese Language Processing, 8(1).

[37] Wu, You-Zheng, J. Zhao, and B. Xu, 2003, “Chinese Named Entity Recognition Combining a Statistical Model with Human Knowledge,” The Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models (ACL 2003), Sapporo, Japan, pp.65-72.

[38] Wu, Xue-Jun, J. B. Zhu, H.Z. Wang, and N. Ye, 2003, “The Application of the Method of Co-Training in Identification of Chinese Organization Names,” The 2003 National Joint Symposium on Computational Linguistics (JSCL-2003)

[39] Xiao, Jing, J. M. Liu, and T. S. Chua, 2002, "Extracting pronunciation-translated names from Chinese texts using bootstrapping approach", Nineteenth International Conference on Computational Linguistics (COLING2002), Taipei, Taiwan, Aug 2002.

[40] Ye, Shi-Ren, T.S. Chua, J. M. Liu, 2002, “An Agent-Based Approach to Chinese Named Entity Recognition”, Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Aug. 2002, pp 1149-1155

[41] Yeh, Ching-Long and H. J. Lee, 1991, “Rule-Based Word Identification for Mandarin Chinese Sentences--a Unification Approach,” Computer Processing of Chinese and Oriental Languages, 5(2), pp. 97-118.

[42] Yu, Shi-Hong, S. H. Bai, and P. Wu, 1998, “Description of the Kent Ridge Digital Labs System Used for MUC-7,” Proceedings of the Seventh Message Understanding Conference (MUC-7).

[43] Zhang, Hua-Ping, Q. Liu, H. Zhang, and X. Q. Cheng, 2002, “Automatic Recognition of Chinese Unknown Words Based on Roles Tagging,” Proceedings of 1st SIGHAN Workshop on Chinese Language Processing.

[44] Zheng, Chen, W. Y. Liu, and F. Zhang, 2002, “A New Statistical Approach to Personal Name Extraction,” ICML 2002, pp. 67-74.

[45] Zipf, George Kingsley, 1932, “Selective Studies and the Principle of Relative Frequency in Language,” Harvard University Press, Cambridge, MA, 1932.

[46] 廈門晚報, 1.24.2003, “廈門人姓名大盤點,”
http://www.csnn.com.cn/csnn0301/ca130860.htm

[47] 袁義達, 2003, “中國姓氏---群體遺傳和人口分佈,” 華東師大

Appendix A　Chinese Named Entity Hierarchy

Code	Category	Description	Note	Lev.	Sta.	Int.	Ext.
I	PER	人名					
I.1	FPN	正式人名					
I.1.a	FPN_CHI	中國人名					
I.1.a.1	CHI_TAI	台灣人名	ex. 陳水扁, 林金塗	2	O	O	O
I.1.a.2	CHI_MC	大陸人名	ex. 胡兵, 陳紅	2	O	O	O
I.1.a.3	CHI_HK	香港人名	ex. 董建華, 梁家輝	2	O	O	O
I.1.a.4	CHI_SM	星馬人名*		2	O	O	O
I.1.b	FPN_JAP	日本人名	ex. 木村拓哉, 野茂英雄	2	O	O	O
I.1.c	FPN_KOR	韓國人名	ex. 李炳圭, 鄭守根	2	O	O	O
I.1.d	FPN_VIE	越南人名*		2	O	O	O
I.2	ALI	別名					
I.2.a	ALI_NIC	暱稱	ex. 小花, 阿土, 來福	2	X	O	O
I.2.b	ALI_SOB	綽號*	ex. 千里眼, 草上飛, 戰神	2	X	X	O
I.2.c	ALI_PSE	字號	ex. 適之, 中山先生, 飲冰室主人	2	X	△	O
I.3	CHA.	擬人物		2	X	△	O
I.4	SPI	神魔精怪		1	X	X	O
I.5	APP.	稱謂	ex. 馬市長, 陳總統				
II	LOC.	地名					
II.1	PLA	地區					
II.1a	PLA_COM	普通地名	ex. 埔頂, 六張犁　Suffix: 地區	1	△	△	O
II.1.b	PLA_CLP	複合地名	ex. 湘西, 嶺南	1	X	O	O
II.1.c	PLA_ADM	行政區劃	Suffix: 洲 國 州 郡 府 城 省 市 縣 鄉 鎮 村 莊 里 寨 區 町 帝國 王國 地方 部落 社區 新村 自治縣 自治洲 自治區 保護區 風景區 共和國 工業區 計畫區 國家公園	0	X	O	X
II.2.	GEO						
II.2.a	GEO_WAT	水域	Suffix: 湖 泊 海 洋 江 河 溪 池 潭 水 埤 泉 井 川 圳 灣 灘 海灘 海岸 海峽 瀑布 海溝 洋流 海子 運河 水庫 溫泉 沙洲 潟湖	0	X	O	X
II.2.b	GEO_TER	地形	Suffix: 山 峰 嶺 崗 岳 島 谷 洞 岬 角 嶼 岩 (山卡) 峽谷 縱谷 鞍部 埡口 走廊 地峽 高原 平原 丘陵 台地 盆地 窪地 火山 山脈 草原 森林 沙漠 大陸 群島 半島 列島 三角洲 沖積扇	0	X	O	X
II.3	ARC.	人工物					
II.3.a	ARC_INF	公共建設	Suffix: 路 街 巷 道 線 關 港 橋 站 公路 鐵路 林道 古道 吊橋 車站 碼頭 漁港 軍港 機場 隧道 公園 花園 農場 牧場 林場 公墓 墓園 果園 營區 大壩 交流道 收費站 休息站 服務區 停車場 產業道路 快速道路 高速公路	1	△	O	O
II.3.b	ARC_BUI	建物	Suffix: 廟 觀 寺 祠 宮 殿 堂 室 樓 館 亭 院 園 閣 軒 齋 陵 碑 塔 巖 大廈 大樓 山莊 禪寺 球場 紀念碑 紀念亭 紀念館 紀念堂 棒球場 足球場 體育館 游泳池 高爾夫球場	2	△	O	X
II.3.c	ARC_FAC	設施	Suffix: 店 分店 門市 國小 國中 高中 女中 大學 中學 小學 高職 醫院 診所 市場 銀行 郵局 農會 飯店 旅社 賓館 民宿 百貨 超市 超商 中心 球館 戲院 影城 樂園 分局 電台 電視 餐廳 派出所 撞球場 研究院 幼稚園 度假村 招待所 加油站 電視台 師範學院 技術學院 活動中心 購物中心 汽車旅館 保齡球館 生鮮超市 黃昏市場	2	X	O	△
II.4	ADD	地址		2	X	O	O
III	ORG	組織團體					
III.1	GOV	行政單位	Suffix: 府 院 部 會 局 司 處 科	0	X	O	X
III.2	GRO	人民團體					
III.2.a	GRO_ASS	社團	Suffix: 會 協會 基金會	2	△	O	X

III.2.b	GRO_PAR	政黨		1	X	O	X
III.2.c	GRO_REL	宗教		1	X	O	X
III.2.d	GRO_TEA	隊伍		2	△	O	△
III.2.e	GRO_GAN	幫會	Suffix: 派 門 幫 教 隊 組	2	X	O	X
III.2.f	GRO_FOR	部隊		1	X	O	△
III.3	RAC	民族		1	O	O	X
III.4	COM	公司行號		2	△	O	△
IV	TERM	術語 (包含學名, 別名)					
IV.1	TAX	生物分類		0	———		
IV.2	CHE.	化學物質		0	———		
IV.3	MED.	醫學名詞		0	———		
IV.4	GES	地球科學		0	———		
IV.5	AST	天文		0	———		
IV.6	PHE.	現象定律理論		0	———		
IV.7	MEA.	度量衡	ex. 伏特, 歐姆	0	———		
IV.8	TSP.	工業型號	ex. 諾克斯級, 飛毛腿型	1	X	O	O
V	TIT	標題					
V.1	PUB.	出版品	Suffix: 網 經 報 日報 晚報 時報 月刊 週刊 週報 畫報 畫刊 雜誌 大全 半月刊 百科全書	1	X	O	O
V.2.	CRE.	創作	ex. 白玉苦瓜, 何日君在來, 教父, 神雕俠侶	2	X	X	O
V.3.	SKI	技藝					
V.3.a	SKI_MAR	武術	ex. 太極拳, 地堂腿	1	X	O	△
V.3.b	SKI_DAN	舞蹈	ex. 爵士, 佛朗明哥	1	△	X	O
V.3.c	SKI_MEL	曲調	ex. 清平樂, 水調歌頭	1	X	X	O
V.3.d	SKI_DIS	陣勢	ex. 長蛇陣, 八陣圖	1	X	O	△
V.4	STY	風格流派	ex. 江西詩派, 淨土宗, 資本主義	1	X	O	△
V.5	CUL	文化	ex. 龍山文化, 繩文時期	1	△	O	O
V.6	BRA	品牌商標	ex. 牛頭牌, 可口可樂	2	X	△	X
VI	MIS	其他					
VI.1	ENT	實物名					
VI.1.a	ENT_VEH	交通工具	ex. 空軍一號, 成功艦	1	X	O	△
VI.1.b	ENT_WEA	武器	ex. 達姆彈, 倚天劍	1	X	O	△
VI.1.c	ENT_FOO	食物餐點	ex. 卡布其諾, 螞蟻上樹	2	X	O	O
VI.1.d	ENT_CLO	衣物布料	ex. 尼龍, 蘇格蘭裙	1	X	O	O
VI.1.e	ENT_OTH	其他		2	———		
VI.2	FES.	節慶	ex. 復活節, 自由日	0	X	O	O
VI.3.	REI	年號	ex. 光緒, 明治	0	O	X	O
VI.4	CON	比賽獎項	ex. 溫布頓公開賽, 諾貝爾獎, 告示牌排行榜	0	X	O	O
VI.5	EVE	歷史事件	ex. 赤壁之戰, 水門案	1	X	O	O
VI.6	PLN	計畫	ex. 沙漠風暴, 黑鷹計畫	2	X	O	O
VII	ABB	簡稱					
VII.1	ABB_SIN	單項簡稱	ex. 皇馬, 僑委會, 劉案, 陳某	3	X	O	△
VII.2	ABB_COM	併稱	ex. 孔孟, 台澎金馬	2	X	△	X
VII.3.	ABB_DER	衍生 NE	ex. 大同貨, 駐美代表	2	△	O	O

Appendix B Surnames and Given Names

Surnames:

Character c	$\ell(SUR\|c)$
陳	1.00
吳	1.00
庾	1.00
郜	1.00
林	0.96
李	0.90
楊	0.86
鄭	0.84
朱	0.77
盧	0.72
胡	0.72
柳	0.61
田	0.53
卞	0.50
葛	0.47
簡	0.46
樊	0.44
祝	0.43
伊	0.42
吉	0.35
屈	0.38
臧	0.33
米	0.30
帥	0.27
沐	0.25
水	0.23
班	0.22
巴	0.20
戎	0.16
秋	0.1
英	0.1

Given names:

Character c	$\ell(GIV\|c)$
文	1.00
岱	1.00
軾	1.00
頫	1.00
美	0.99
惠	0.98
玉	0.96
婷	0.93
華	0.92
志	0.91
家	0.87
國	0.84
仁	0.82
傑	0.79
姿	0.76
雄	0.74
達	0.68
筱	0.65
全	0.64
婉	0.62
大	0.55
旺	0.53
玨	0.50
河	0.43
本	0.39
嵩	0.33
厚	0.25
箏	0.25
塗	0.20
霄	0.14
固	0.1
直	0.1

Appendix C　Location Suffixes and Organization Suffixes

Location Suffixes:

洲 國 州 郡 府 城 省 市 縣 鄉 鎮 村 莊 里 寨 區 町 帝國 王國 地方 部落 社區 新村 自治縣 自治洲 自治區 保護區 風景區 共和國 工業區 計畫區 國家公園 湖 泊 海 洋 江 河 溪 池 潭 水 埤 泉 井 川 圳 灣 灘 海灘 海岸 海峽 瀑布 海溝 洋流 海子 運河 水庫 溫泉 沙洲 潟湖 山 峰 嶺 崗 岳 島 谷 洞 岬 角 嶼 岩 峽谷 縱谷 鞍部 埡口 走廊 地峽 高原 平原 丘陵 台地 盆地 窪地 火山 山脈 草原 森林 沙漠 大陸 群島 半島 列島 三角洲 沖積扇 路 街 巷 道 線 關 港 橋 站 公路 鐵路 林道 古道 吊橋 車站 碼頭 漁港 軍港 機場 隧道 公園 花園 農場 牧場 林場 公墓 墓園 果園 營區 大壩 交流道 收費站 休息站 服務區 停車場 產業道路 快速道路 高速公路 廟 觀 寺 祠 宮 殿 堂 室 樓 館 亭 院 園 閣 軒 齋 陵 碑 塔 巖 大廈 大樓 山莊 禪寺 球場 紀念碑 紀念亭 紀念館 紀念堂 棒球場 足球場 體育館 游泳池 高爾夫球場店 分店 門市 國小 國中 高中 女中 大學 中學 小學 高職 醫院 診所 市場 銀行 郵局 農會 飯店 旅社 賓館 民宿 百貨 超市 超商 中心 球館 戲院 影城 樂園 分局 電台 電視 餐廳 派出所 撞球場 研究院 幼稚園 度假村 招待所 加油站 電視台 師範學院 技術學院 活動中心 購物中心 汽車旅館 保齡球館 生鮮超市 黃昏市場 人 族

Organization Suffixes:

Typing Suffixes:

會 協會 基金會 工會 同業公會 宗親會 聯誼會 校友會 公司 集團 有限公司 股份有限公司 機構 工作坊 商行 會社 行 社 派 門 幫 教 隊 組

Business Descriptions:

木工機械 模具制品 形象廣告 汽車修理 食品醬菜 鋁質電鍍 行銷企劃 五金制品 吊車搬家 音響視聽 電子材料 特殊鋼鐵 影音系統 移民留學 機車託運 塑膠製品 絕緣材料 健康商品 電話消毒 行銷管理 資訊技術 影視設備 工業電腦 木製品 保險櫃 電光源 藝術品 生產力 安全鞋 整流器 控制閥 百葉窗 鐵絲網貨柜場 領導學 儲水桶 淨水器 混凝土 帆布業 禮藝品 貨櫃 航運 禮服 時裝 文化 攝影 音響 電器 特機 藤業 文具 鎖鍊 帽子 冷氣 水電 毛巾 膠業 乾冰 針車 拍賣 油業 電話 化工 化學 水泥 瓦斯 生技 生活 石化 交通 水床 洗衣 有機 文教 托運 財務 快譯 角鐵 快遞 餐飲 影音 圖書 鋼索 製帽 期貨 蜜餞石藝

Affair Descriptions:

進出口 包工業 工業 企業 事業 實業 興業 展業 商事 工程 科技 開發 研發 顧問 建設 製造 信託 投資 金融 貿易 發展 連鎖 商業 工貿 商貿洋行 仲介 產業 經紀 招標 經理 流通 服務 承攬 出版 代理

Appendix D　Often Seen Monosyllabic Words

爲的不於向已是有去最就到會在了當應較自並很大中小至受被把將拿幫替像待朝望問對要以能可般似往與除從連同和用藉憑仗假因據視隨繼等趁趕距離打逢隔迄上經靠俟下前後內外裡旁東西南北底末初邊方側端來時該須得別遭跟休甭勿免依按照比如擬未你我他妳她也而指使僅本抓仍寫卻說做隊若或曾

Appendix E　Sample Testing Sentences

Statistical Deviations:

游揆原 希望 林嘉誠 留 在 內閣
下台 的 余政憲 與 簡 又 新 形塑 出 主動 爲 政策 負責 姿態
宋再以 查 無 証據 簽 結
美國 副總統 錢 尼 訪問 大陸

Misleading Suffixes:

現任 政次 楊子 江有銀行 背景
他 也 不會 出任 台銀 董事 長和開發基金 執行秘書
黑道派 小弟 當他 司機
目前 政府 尚未 開放 大 煉鋼廠 到 大陸投資
國壽投資 中華電信 是 基於 長期 財務 收入 著眼
同時 並將 發 展爲控股公司 的 模式
季後 賽大舞台燈光 一 打開
姚元浩站 在 女孩 們 身後 擺 V 手勢
郭富城教 大家 一起 擺出 飛翔 的 動作

OOV:

個人 的 第二發 沙 喲 娜娜安
把 壘 上三名 隊友 全部 送回 本壘

Insufficiencies of Handcrafted Knowledge:

分別 是 國科會 主委 魏哲和 和海 巡 署長 王郡

包括 考選部 、 銓敘部 、 保 訓 會正 副首長 與 秘書長 朱武獻 等 七位 政務官 已 遞出 辭呈

基層 教師 協會 、 快樂 學習 連線 昨天 要求 杜正勝 先 檢討 教育 經費分配

嘉義縣 野鳥 協會 總幹事 葉明宗 、 資深 會員 翁榮炫 等 人

一處 鄰近 卑南 史前文 化公園 旁 的 釋迦 果園 整地

中央研究院 歷史 語言 研究所 、 國立 台灣大學 、 史前館 等 單位 過去 都曾 參與 過 卑南 史前 遺址 發掘 研究

約 五十年 歷史 的 代天府 主要 供奉 五府 千歲

與 寧波 北崙經濟技術 特區 僅 一 水之 隔

La new 熊周森毅 第二局 觸擊 球 落在 本壘 前

紐約 大都會 隊 日籍 球員 松 井 稼 頭 央首局 首 打 席 就 轟出 全壘打

澎 恰恰 表示 很 感謝 劉 真的 幫忙

Shortcomings of Heuristic Rules

游揆原 希望 林嘉誠 留 在 內閣

不管 是 能力 與 資 望都 在 黨內 足以 服眾

即 把 戳記 蓋在 投票 格內 及 姓名 、 號碼 上 的 選票

應 先和 主計 單位 協調

一直 採 「 只 許成功 」 的 高 規格 原則

就 常 跟 台南 市警 界人士 出入 有 粉味 的 酒店

這 就是 成長一 部份

在 布袋 鎮西濱公路 接近 新岑國 小 附近 廢棄 鹽灘地

就 連 喝 咖啡 的 錢都 能 換算成 回饋金

九局 再 下一城

另外 還有 一位 國 小劉 姓 同學 揭發 他 從小 就 「 娘 」 得 很

Inherent Ambiguity

意外 發現 股東 之一 的 黃村曾 「打死 人 卻 沒事」

賈奈 特 耐 撞

姚明初 嘗 季後賽 滋味

只能 乾瞪眼 看 金塊 採花 安東尼大 顯身手