A Unicode based Adaptive Segmentor

Q. Lu, S. T. Chan, R. F. Xu, T. S. Chiu
Dept. Of Computing,
The Hong Kong Polytechnic University,
Hung Hom, Hong Kong
{csluqin,csrfxu}@comp.polyu.edu.hk

B. L. Li, S. W. Yu
The Institute of Computational Linguistics,
Peking University,
Beijing, China
{libi,yusw}@pku.edu.cn

Abstract

This paper presents a Unicode based Chinese word segmentor. It can handle Chinese text in Simplified, Traditional, or mixed mode. The system uses the strategy of divide-and-conquer to handle the recognition of personal names, numbers, time and numerical values, etc in the pre-processing stage. The segmentor further uses tagging information to work on disambiguation. Adopting a modular design approach, different functional parts are separately implemented using different modules and each module tackles one problem at a time providing more flexibility and extensibility. Results show that with added pre-processing modules and accessorial modules, the accuracy of the segmentor is increased and the system is easily adaptive to different applications.

1 Introduction

The most difficult problem in Chinese word segmentation is due to overlapping ambiguities [1-2]. The recognition of names, foreign names, and organizations are quite unique for Chinese. Some systems can already achieve very high accuracy [3], but they heavily rely on manual work in getting the system to be trained to work certain language environment. However, for many applications, we need to look at the cost to achieve high accuracy. In a competitive environment, we also need to have systems that are quickly adaptive to new requirements with limited resources available.

In this paper, we report a Unicode based Chinese word segmentor. The segmentor can handle Chinese text in Simplified, Traditional, or mixed mode where internally only one dictionary is needed. The system uses the strategy of divide-and-conquer to handle the recognition of personal names, numbers, time and numerical values. The system has a built-in new word extractor that can extract new words from running text, thus save time on training and getting the system quickly adaptive to new language environment. The Bakeoff results in the open text for our system in all categories have shown that it works reasonably good for all different corpora.

The rest of the paper is organized as follows. Section 2 presents our system design objectives and components. Section 3 discusses more implementation details. Section 4 gives some performance evaluations. Section 5 is the conclusion.

2 Design Objectives and Components

With the wide use of Unicode based operating systems such as Window 2000 and Window XP, we now see more and more text data written in both the Simplified form and the Traditional form to co-exist on the same system. It is also likely that text written in mixed mode. Because of this reality, the first design objective of this system is its ability to handle the segmentation of Chinese text written in either Simplified Chinese, Traditional Chinese, or mixed mode. As an example, we should be able to segment the same sentence in different forms such as the example given below:

```
Simplified:[我][后日][要][上学]
Traditional:[我][後日][要][上學]
Mixed:[我][后日][要][上學]
```

The second design objective is to adopt the modular design approach where different functional parts are separately implemented using independent modules and each module tackles one problem at a time. Using this modular approach, we can isolate problems and fine tune each module with minimal effect on other modules in the system.

Special features like adding new rules or new dictionary can be easily done without affecting other modules. Consequently, the system is more flexible and can be easily extended.

The third design objective of the system is to make the segmentor adaptive to different application domains. We consider it having more practical value if the segmentor can be easily trained using some semi-automatic process to work in different domains and work well for text with different regional variations. We consider it essential that the segmentor has tools to help it to obtain regional related information quickly even if annotated corpora are not available. For instance, when it runs text from Hong Kong, it must be able to recognize the personal names such as 陳方安生 if such a name(quadra-gram) appears in the text often.

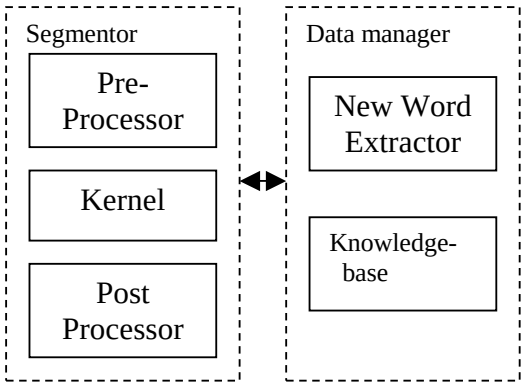

Figure 1. System components

Figure 1 shows the two major components, the segmentor and data manager. The *segmentor* is the core component of the system. It has a pre-processor, the kernel, and a post-processor. As the system has to maintain a number of tables such as the dictionaries, family name list, etc., a separate component called data manager is responsible in handling the maintenance of these data. The *pre-processor* has separate modules to handle paragraphs, ASCII code, numbers, time, and proper names including personal names, place and organizational names, and foreign names. The *kernel* supports different segmentation algorithms. It is the application or user's choice to invoke the preferred segmentation algorithms that at current time include the basic maximum matching and minimum matching in both forward and backward mode. These can also be used to build more complicated algorithms later on. In addition, the system provides segmentation using part-of-speech tagging information to help resolve ambiguity. The *post-processor* applies morphological rules which cannot be easily applied using a dictionary.

The data manager helps to maintain the knowledge base in the system. It also has an accessory software called the *new word extractor* which can collect statistical information based on character bi-grams, tri-grams and quadra-grams to semi-automatically extract words and names so that they can be used by the segmentor to improve performance especially when switching to a new domain. Another characteristic of this segmentor is that it provides tagging information for segmented text. The tagging information can be optionally omitted if not needed by an application.

3 Implementation Details

The basic dictionary of this system was provided by Peking University [4] and we also used the tagging data from [4]. The data structure for our dictionaries are very similar to that discussed in [5]. As our program needs to handle both Simplified and Traditional Chinese characters, Unicode is the only solution for dealing with more than one script at the same time.

Even though it is our design objective to support both Simplified and Traditional Chinese, we do not want to keep two different sets of dictionaries for Simplified and Traditional Chinese. Even if two versions are kept, it would not serve well for text in mixed mode. For example, Traditional Chinese word of "the day after tomorrow" should be 後日, and for Simplified Chinese, it should be 后日. However sometimes we can see the word 后日 appears in a Traditional Chinese text. We cannot say that it is wrong because the sentence is still semantically correct especially in Unicode environment. Therefore the segmentor should be able to segment those words correctly such as in the examples: "[我][後日][上班]", and in "[他][后日][放假]". We must also deal with dictionary maintenance related to Chinese variants. For example, characters 強 and 强 are variants, so are 鵬 and 鹏.

In order to keep the dictionary maintenance simple, our system uses a single dictionary which only keeps the so called canonical form of a word. In our system, the *canonical form* of a word is its "simplified form". We quoted the word "simplified" because only certain characters have simplified forms such as 後日 to 后日, but for 清晨, there is no simplified form. In the case of variants, we simply choose one of them as the canonical character. The canonical characters are maintained in the traditional-simplified character conversion table as well as in a variant table. Whenever a new word, *item*, is added into the dictionary, it must be added using a function *CanonicalConversion()*, which takes *item* as an input. During segmentation, the corresponding dictionary look up function will first convert the token to its canonical form before looking up in the dictionary.

The personal name recognizers (separate for Chinese names and foreign names) use the maximum-likelihood algorithm with consideration of commonly used Chinese family names, given names, and foreign name characters. It works for Chinese names of length up to 5 characters. In the following examples you can see that our system successfully recognized the name 陳方安生. This is done using our algorithm, not by putting her name in our dictionary:

[陳方安生/nr][昨晚/t][出席/v][公開/a][場合/n][發表/v][演詞/n]

Organization names and place names are recognized mainly using special purpose dictionaries. The segmentor uses tagging information to help resolve ambiguity. The disambiguation is mostly based on rules such as

p + (n + f) -> p + n + f

which would word to correct

[從/p][(馬/n 上/f)/d] -> [從/p][馬/n][上/f]

For efficiency reasons, our system uses only about 20 rules. The system is flexible enough for new rules to be added to improve performance.

The new word extractor is an accessory program to extract new words from running text based on statistical data which can either be grabbed from the internet or collected from other sources. The basic statistical data include bi-gram frequency, tri-gram frequency, and quadra-gram frequencies. In order to further example whether a bi-gram, say 差餉, is indeed a word, we further collect forward conditional frequency of 差, $freq_{forward}(餉|差)$, and the back-ward conditional frequency of 餉, $freq_{backward}(差|餉)$. For an i-gram token, we also use the (i+1)-gram statistics to eliminate those i-grams that are only a part of (i+1) – gram word. For instance, if the frequency of bi-gram 普洱 is very close to the frequency of tri-gram 普洱茶, it is less likely that 普洱 is a word. Of course, whether 普洱茶 is a word depends on quadra-gram results. Using the statistical result, a set of rules was applied to these i-grams to eliminate entries that are not considered new words. Minimal manual work is required to identify whether the remaining candidates are new words. Before words are added into the dictionary, part-of-speech information are added manually (although not necessary) before using the canonical function. The following table shows examples of bi-grams which are found by the new word extractor using one year Hong Kong Commercial Daily News data.

Bigram	count	forwardfreq	backwardfreq
差餉	818	0.061425	0.943483
入伙	734	0.007373	0.583002
叱吒	120	0.902256	0.794702
蜆殼	106	0.751773	0.120045
叮噹	44	0.330827	0.473118
普洱	44	0.003137	0.483516
變重	15	1.000000	0.001847
荷蘭	15	1.000000	1.000000

4 Performance Evaluation

The valuation metrics used in [6] were adopted here.

$$recall = \frac{N_3}{N_1} \quad (1)$$

$$presicion = \frac{N_3}{N_2} \quad (2)$$

$$F_1(precision, recall) = \frac{2 \times recall \times precision}{recall + precision} \quad (3)$$

where N_1 denotes the number of words in the annotated corpus, N_2 denotes the number of words identified by the segmentation algorithm , and N_3 is the number of words correctly identified.

We participated in the open tests for all four corpora. The results are shown in the following table.

Data	R	P	F	R_{oov}	R_{iv}
AS	0.892	0.853	0.872	0.236	0.906
CTB	0.853	0.806	0.829	0.578	0.914
HK	0.909	0.863	0.886	0.579	0.935
PK	0.94	0.911	0.925	0.647	0.962

The worst performance in the 4 tests were for the CTB(UPenn) data. From the observation from the testing data, we found that the main problem with have with CTB data is the difference in word granularity. To confirm our observation, we have done an analysis of combining errors and overlapping errors. The results show that the ratios of combining errors in all the error types are 0.8425(AS), 0.87684(CTB), 0.82085(HK), and 0.77102(PK). The biggest problem we have with AS data, on the other hand is due to out of vocabulary mistakes. Even though our new word extractor can help us to reduce this problem, but we have not trained our system using data from Taiwan. Our best performance was on PK data because we used a very similar dictionary. The additional training of data for HK was done using one year Commercial Daily(香港商報).

The following table summarizes the execution speed of our program for the 4 different sources:

Data	No. of chars	Processing Time (sec.)	Processing Rate (char/sec)	Segmentation Rate (char/sec)
AS	18,743	4.703	3,985	7,641
CTB	62,332	10.110	6,165	7,930
HK	57,432	10.329	5,560	7,109
PK	28,458	4.829	5,893	10,970

The program initialization needs around 2.25 seconds mainly to load the dictionaries and other data into the memory before the segmentation can start. If we only count the segmentation time, the rate of segmentation on the average is around 7,500 characters for the first three corpora. It seems that the processing speed for Peking U. data is faster. This may be because the dictionaries we used are closer to the PK system, thus it would take less time to work on disambiguation.

5 Conclusion

In this paper, design and algorithms of a general-purposed Unicode based segmentor is proposed. It is able to process Simplified and Traditional Chinese appear in the same text. Sophisticated pre-processing and other auxiliary modules help segmenting text more accurately. User interactions and modules can be easily added with the help of its modular design. A built-in new word extractor is also implemented for extracting new words from running text. It saves much time on training and thus it can be quickly adapted to new environments.

Acknowledgement

We thank the PI of ***ITF Grant by ITC of HKSARG*** (ITS/024/01) entitled: *Towards Cost-Effective E-business in the News Media & Publishing Industry* for the use of HK Commercial Daily.

References

[1] Automatic Segmentation and Tagging for Chinese Text (《中文文本自動分詞和標注》) , K.Y. Liu, Commercial Press, 2000

[2] Segmentation Issues in Chinese Information Processing, 《語言文字應用》(C.N. Huang Issue No. 1, 1997)

[3] The design and Implementation of a Modern General Purpose Segmentation System (B. Lou, R. Song, W.L. Li, and Z.Y. Luo, Journal of Chinese Information Processing, Issue No. 5, 2001)

[4] 北大語言資源：語法資訊辭典 (Institute of Computational Linguistics, Peking Univ., 2002)

[5] 漢語自動分詞詞典機制的實驗研究 (孫茂松, 左正平, 黃昌寧, Journal of Chinese information processing vol. 14, no. 1, 2001)

[6] Chinese Word Segmentation and Information Retrieval, Palmer D., and Burger J., In AAAI Symposium Cross-Language Text and Speech Retrieval 1997