Chinese Word Segmentation with a Maximum Entropy Approach

Low Jin Kiat

(B.Computing.(Computer Science), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2006

Acknowledgements

I thank my thesis supervisor and mentor, A/P Ng Hwee Tou, for his guidance and support throughout the project. I have benefitted greatly from his insights and visions. His valuable advice and encouragements have been a great help to the completion of this project.

I thank my colleague Guo Wenyuan from the Computational Linguistics Lab for his assistance during the participation of the Sighan Bakeoff 2, and the helpful comments he gave for this thesis.

I like to thank my colleagues in the Computational Linguistics Lab for their friendship and support.

Finally, I would like to thank my family for their support and encouragement during my studies.

Table of Contents

Summary

In this thesis, we present a maximum entropy approach to Chinese word segmentation. Besides using features derived from gold-standard word-segmented training data, we also used an external dictionary and additional training corpora of different segmentation standards to further improve segmentation accuracy. The selection of useful additional training data is modeled as example selection from noisy data. Using these techniques, our word segmenter achieved state-of-the-art accuracy. We participated in the Second International Chinese Word Segmentation Bakeoff organized by SIGHAN, and evaluated our word segmenter on all four test corpora in the open track. Among 52 entries in the open track, our word segmenter achieved the highest F measure on 3 of the 4 test corpora, and the second highest F measure on the fourth test corpus.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Chinese Word Segmentation Problem

The fact that Chinese texts come in an unsegmented form causes problems for applications which require the input text to be segmented into words. Before we can carry out more complex Natural Language Processing (NLP) tasks like machine translation and text-to-speech synthesis, Chinese word segmentation is a necessary first step. Even though a Chinese text is made up of words, the word boundaries are not explicitly marked in Chinese. A Chinese text is written as a continuous string of characters without any intervening space, and words are not demarcated. Each character can be a word by itself, or can be part of a larger word which is made up of two or more characters. To illustrate, consider the Chinese character "草" (grass) which can be a single word. It can also be the second character in a two character word "潦草" (sloppy, untidy), or the first character in the word "草芥" (trifle, insignificant). To determine where the word boundary should be placed for a word, we need to consider the surrounding context.

Furthermore, the interpretation of a sentence also changes when a text is

segmented in different ways. Consider the following example:

"我到超市买新西兰花。"[1]

This sentence could essentially translate into two correct though different interpretations under two different segmentations although (a) is more likely given the context:

a) "我 到 超市 买 新 西兰花 。"

I went to the supermarket to buy fresh broccoli.

b) "我 到 超市 买 新西兰 花 。"

I went to the supermarket to buy New Zealand flowers.

Therefore, producing an accurate word segmenter is important, since the meaning of a sentence can change as a result of assigning a different segmentation. However, Chinese word segmentation is not a trivial task as a result of the segmentation ambiguity of characters. The surrounding context of a character is particularly important in determining the correct segmentation.

Another major challenge in Chinese word segmentation is the correct segmentation of unknown, out-of-vocabulary (OOV) words. Though the number of characters in the Chinese language is relatively constant, this is not true for words. New out-of-vocabulary words cause significant accuracy degradation in Chinese word segmentation. In the first SIGHAN International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003), results of the participants in the closed category strongly indicate that OOV words have a strong impact on the segmentation accuracy. Accuracy on a test corpus like the AS test corpus which has a low OOV rate of 2.2% was significantly higher than the

[1]Adapted from Teahan *et al.* (2000)

other test corpora, such as CTB which has a high 18.1% OOV rate. Therefore effectively identifying new words is important in achieving a high word segmentation accuracy. But it is not possible to provide dictionaries or training corpora that include all words since new words appear constantly. This could be due to new person names (a new Chinese name may be formed by a different combination of Chinese characters), new technical terms, or transliterations of new English terms. Moreover, dictionaries do not provide the necessary context for a word, and as we have previously seen, the same sequence of characters can have different segmentations based on the context.

1.2 Applications of Chinese Word Segmentation

Chinese word segmentation is a necessary pre-requisite for many NLP tasks. Characters by themselves can appear with different meanings in different context, and it is only in word-segmented form that a sentence can be meaningful enough to be processed by computer systems for various NLP tasks like machine translation, named entity recognition, and speech-to-text synthesis. We present a few key areas in which word segmentation is required as a pre-processing task.

1.2.1 Machine Translation

Machine translation relies on the concept of a "word". In order to correctly translate a Chinese sentence into English, the Chinese sentence has to be correctly segmented into words first before translation. It is only with correct and accurate word segmentation that a sentence can have a correct transla-

tion. A wrong translation can be intolerable since each translation can convey drastically different meaning.

1.2.2 Digital Library Systems

Chinese word segmentation forms an important component of a Chinese digital library system. With the huge amount of text that is present in a digital library, full-text indexing is almost a must for any digital library system. Techniques based on full-text indexing were developed using languages like English in which word boundaries are given. If text indexing was built from characters rather than words, then searches will suffer from the problem of low precision, with many irrelevant documents being returned, since characters can be used in many different contexts different from that of the intended query. Similarly, in information retrieval systems, the relevance of a document to a query relies on term frequency of words. A document is ranked higher if it contains more occurrences of the query terms. The relationship between the frequency of a word and a character that appears within the word is weak. Hence without word segmentation, the precision of a search will be lower since relevant documents would be less likely to be ranked high in the search. For example, the component characters " 草" and " 原"of the word " 草原"(grassland) can appear in many different words such as "原来"(original), "草席"(straw mat), and "原谅"(forgive), which have different meanings from the component characters. A study conducted by Broglio *et al.* (1996) concludes that the performance of an unsegmented character based query is about 10% lower than that of the corresponding segmented query. An accurate word segmenter would therefore help the many applications in digital library systems such as text retrieval, text summarization and document clustering.

1.3 Contributions

In this thesis, we present a machine learning approach for accurate Chinese word segmentation. Our basic approach is based on maximum entropy modeling. Through the introduction of appropriate and useful features, we sought to create a flexible and accurate segmenter that is able to segment Chinese text accurately according to the required segmentation standard. In order to deal with the OOV problem, we also sought to incorporate additional dictionary features based on an external word list, and to use extra training data annotated in other word segmentation standards. Corpora of different segmentation standards are able to provide a rich source of knowledge, with the necessary context features. Effectively, we are pooling the relevant and useful knowledge resources across corpora of different segmentation standards for use in training a word segmenter. In this thesis, we selected the relevant extra training samples by removing the potentially noisy, wrongly segmented characters. As far as we know, this is the first work in Chinese word segmentation that attempts to incorporate useful extra training data from different segmentation standards for use in training a segmenter automatically.

We carried out comprehensive experiments on all 8 datasets from the First and Second International Chinese Word Segmentation Bakeoff and obtained state-of-the-art results on all 8 datasets. In general, the use of an external dictionary and corpora of different segmentation standards to supplement the existing training data have provided consistent improvements over the use of just basic features.

1.4 Organization of the Thesis

The structure of this thesis is as follows: In Chapter 2, we review Chinese word segmentation research. Chapter 3 provides some basic theory of maximum entropy modeling and two parameter estimation algorithms: GIS and LBFGS. In Chapter 4, we describe our basic word segmentation method and the basic set of features we employed. Then in Chapter 5, we address the problem of OOV words through two proposed methods: use of dictionary features, and selection of extra training data from corpora of different segmentation standards. In Chapter 6, we provide a comprehensive evaluation of the performance of our word segmenter when tested on the first and second SIGHAN bakeoff datasets. We conclude in Chapter 7 and suggest some possible future work.

Chapter 2

Approaches to Chinese Word Segmentation

In this chapter, we review related research on Chinese word segmentation. Popular methods include dictionary-based methods, statistics based methods, and their combination. We also review the machine learning, corpus-based approach to Chinese word segmentation, a popular approach in recent times.

Though there was not as much morphological research on Chinese compared to English morphological work, Chinese morphological research is now gaining a higher level of interest from the research community, with the availability of data and the growth of the Chinese language as one of the most commonly used online languages on the Internet. Most of the Chinese word segmentation systems reported previously can generally be classified into three main approaches:

1) Dictionary-based methods, with some grammar rules to resolve ambiguities.

2) Statistics based methods, using statistical counts of characters in a training corpus to estimate probability;

3) Combination of both

2.1 Dictionary-Based Methods

Dictionary-based approaches (Chen and Liu, 1992; Cheng *et al.*, 2003) involve the use of a machine-readable dictionary (word list) independent of the test set, and grammer rules to deal with segmentation ambiguities. The most common method to deal with ambiguities in word segmentation in this approach is the maximum matching algorithm. Different variants of the algorithm exist, the most basic one being the "greedy" version, which finds the longest word (from the dictionary) starting from a character and then continuing on with the next character till the whole sentence is processed. For example, given that the words "东" (east), "西" (west), and "东西" (thing) are found in the dictionary, the greedy algorithm will choose "东西" as the word if it encounters a sequence of characters "东西" in the sentence. Though simple, it has been empirically found to be able to achieve over 90% segmentation accuracy if the dictionary is large. However in reality, no dictionary is complete with all possible words and it would probably be unrealistic to apply a pure dictionary-based method for segmentation. The strength of a dictionary-based approach lies in its simplicity and efficiency. But with computing resources being able to handle more computationally intensive work required for machine-learning, corpus-based approach, the trend is now moving towards machine-learning approaches.

2.2 Statistics-Based Methods

Statistical approaches include that from Sproat and Shih (1990). Their approach focuses on two-character words and uses the mutual information of two adjacent characters to decide if they should form a word. Adjacent characters in a sentence with the largest mutual information above a set threshold would be grouped together as a word. Another statistical approach of Dai *et al.* (1999) also considers two-character words. In their work, they explored different notions of frequency of bigrams and characters, including relative frequency, weighted document frequency, and document frequency. In their work, they found contextual information to be one of the most useful features in determining a word boundary. Like the dictionary based approach, the statistics-based approach is simple and efficient, but accuracy wise, it is not as high as a machine learning, corpus based approach.

2.3 Hybrid Methods

Hybrid approaches combine the use of dictionary and statistical information for word segmentation. Compared with purely statistical approaches, hybrid approaches have the guidance of a dictionary and as a result they generally outperform statistical approaches in terms of segmentation accuracy. As an example, Sproat *et al.* (1997) introduce a hybrid based approach. They view Chinese word segmentation as a stochastic transduction problem, and introduce a zeroth-order language model for Chinese word segmentation, and finding the lowest summed unigram cost in their model. Each word in the dictionary is represented as a sequence of arcs, each labeled with a Chinese character and its Chinese pinyin syllables, starting from an initial state and terminated by

a weighted arc labeled with an empty string ε and a part-of-speech tag. The weight represents the estimated cost of the word, and the best segmentation is taken to be the path that has the cheapest cost for the sequence of characters in the sentence.

2.4 Supervised Machine Learning Methods

More recent and more successful studies in the field would involve some form of supervised machine learning approaches (Luo, 2003; Ng and Low, 2004; Peng *et al.*, 2004; Xue and Shen, 2003). Luo (2003), Xue and Shen (2003), and Ng and Low (2004) make use of a maximum entropy (ME) modeling approach to perform Chinese word segmentation. In their work, four possible classes (or tags) were used for each character to denote the relative position of the character within a word: one tag for a character that begins a word, and is followed by another character; another tag for a character that occurs in the middle of a word; another tag for a character that ends a word; and another tag for a character that occurs as a single-character word. This is similar to using chunk-based tags as classes in base noun-phrase chunking (Erik *et al.*, 2000). Peng *et al.* (2004) applied Conditional Random Fields(CRFs) modeling for Chinese word segmentation and like the above mentioned works, made use of the character context features and external dictionary in segmentation. However, Peng *et al.* (2004) only used two possible classes (or tags) to denote if a character starts a word or ends a word, and also included a separate OOV detection phase to detect OOV words in the test data.The success of the ME model largely depends on selecting the appropriate features to aid in classification. For the Chinese word segmentation task, common features like single characters, combination of adjacent characters were used.

Goh *et al.* (2004) introduced a combined dictionary-based approach with machine-learning in their word segmenter. Like Xue and Shen (2003), each character is assigned one of four possible word boundary tags. In their proposed method, the forward maximal matching (FMM) algorithm and backward maximal matching (BMM) algorithm are first applied to the unsegmented text. Both algorithms match the longest word (from the dictionary) starting from a character (the two algorithms differ in which end of the sentence is the starting character and the direction of movement). Based on the results of the FMM and BMM algorithm and the context of the characters, a Support Vector Machine (SVM) classifier is then used to reassign the word boundaries. SVMs classify data by mapping it into a high dimensional space and constructing a maximum margin hyperplane to separate the classes in the space. Another related work is that from Gao *et al.* (2004) who approached the Chinese word segmentation problem using linear models and Transformation-Based Learning (TBL). Gao *et al.* (2004) used a large MSR corpus, comprising of about 20 million words as their main training data source to train their segmenter. Then standard adaptation is conducted by a TBL postprocessor which performs a set of transformation on the output of the original segmenter in order to obtain the new segmentation standard required. Supervised learning approaches like maximum entropy and SVM allow the flexibility of incorporating contextual information as features in the modeling process. In the supervised learning approach, useful and important features need to be identified for the task. The supervised machine learning approach has been found to give high accuracy, and in the recent second SIGHAN bakeoff, top systems in the open and closed category such as (Asahara *et al.*, 2005; Low *et al.*, 2005; Tseng *et al.*, 2005) have all successfully adopted a machine learning approach to Chinese word

segmentation.

Chapter 3

Basic System overview

In this chapter, we present our basic approach to the Chinese word segmentation problem and introduce maximum entropy (ME) modeling as our main modeling technique to solving the Chinese word segmentation problem. We also briefly review two popular parameter estimation algorithms for maximum entropy, Generalized Iterative Scaling (GIS) and metric variable methods (LBFGS).

3.1 Supervised, Corpus-Based Approach

Our work follows a machine-learning, corpus-based approach. In this approach, we make use of a training set which is a large set of training examples, annotated with the correct classes for which we are interested in finding. With this large annotated training material, we extract the relevant features for each training example, and form the relevant training vectors. We would then use these training feature vectors to train a classifier, which would be able to predict the class when given a new test example. Thus, once training has been done with a correctly hand annotated corpus, the task would then be to find

the most probable class to assign to each testing example. To summarize, this supervised machine-learning, corpus based approach consists of three main processes: feature extraction, classifier training, and classifier prediction for a test example. The general process is shown in Figure 3.1. The choice and quality of the training corpus and the training algorithm, plus the features chosen for a particular task has a big influence on the accuracy of the classifier. The training corpus used for our work comes from the official SIGHAN bakeoffs, all with varying quantity and vocabulary coverage. For the classifier training algorithm, we chose GIS or LBFGS as the main algorithm for training the maximum entropy classifier. Maximum entropy modeling has been successfully applied in many NLP applications with great success.

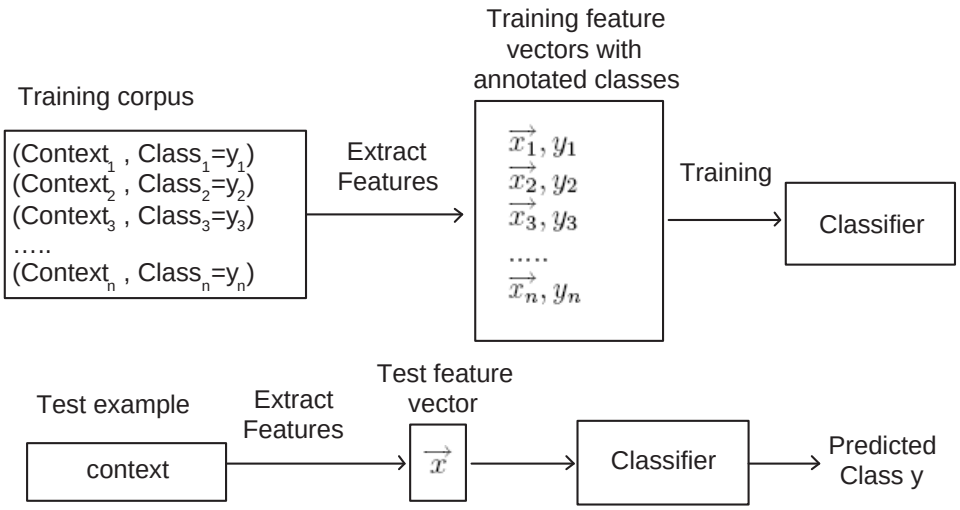

Figure 3.1: General Overview of a Machine-Learning, Corpus-Based Approach

3.2 Maximum Entropy Modeling

Chinese word segmentation can be formulated as a statistical classification problem, in which the task is to estimate the "class c" occurring with the highest probability given a "history h" (context). The training corpus usually contains information which suggests the relation between "class c" and "history h", but never enough to specify $p(c|h)$ for all possible (c, h) pairs. The principle of maximum entropy states that in making inferences in the presence of partial information, in order not to make arbitrary assumptions which are not warranted, the probability distribution function has to have the maximum entropy. In this thesis, our word segmenter is built using a maximum entropy framework. The maximum entropy framework has been successfully applied in many NLP tasks (Chieu and Ng, 2002; Ratnaparkhi, 1996; Xue and Shen, 2003), achieving high accuracy when compared with other machine learning approaches. It is based on maximizing the entropy of a distribution subject to the constraints derived from the training data, which link aspects of what we observe with an outcome class that we wish to predict. The probability distribution has the form (Pietra *et al.*, 1997):

$$P(c|h) = \frac{1}{Z(h)} \prod_{j=1}^{k} \alpha_j^{f_j(h,c)}$$

where c is the outcome class, h is the history (context) observed, $Z(h)$ is a normalization constant, $f_j(h, c) \in \{0, 1\}$, and α_j is a "weight" corresponding to feature f_j . There exist a number of algorithms for estimating the parameters of ME models, including iterative scaling, gradient ascent, conjugate gradient, and variable metric methods. One of the more commonly used algorithms is the standard Generalized Iterative Scaling (GIS) (Darroch and Ratcliff, 1972) method, which improves the estimation of the parameters at each iteration.

However, some recently published results (Malouf, 2002) have suggested that the limited memory variable metric algorithm (LBFGS) is better than the GIS algorithm in estimating the maximum entropy model's parameters for the NLP tasks they have tested on. We conducted a series of experiments to compare the accuracy obtained from these two different parameter estimation algorithms. Based on our findings on the Chinese word segmentation task using bakeoff 1 and 2 data, we found LBFGS to perform slightly better than GIS, though LBFGS requires more iterations to converge and longer training time for this task. Our final word segmenter was built using LBFGS as the parameter estimation algorithm.

Figure 3.2 shows a system overview of how we conduct training and testing using the maximum entropy approach.

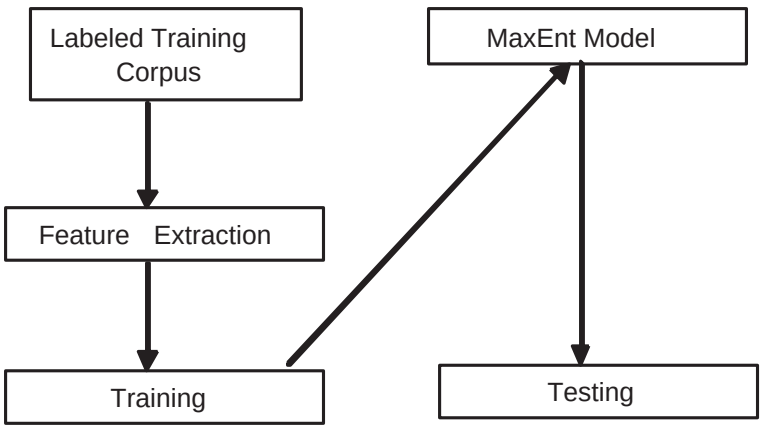

Figure 3.2: Basic System Overview

3.2.1 Parameter Estimation Algorithms

Our presentation of the parameter estimation algorithms follows that of (Wallach, 2002).

Generalized Iterative Scaling

Generalized iterative scaling seeks to improve the log-likelihood of the training data in an incremental manner. Recall that in the maximum entropy framework, we have a classification model $p(y|x, \Theta)$, parameterized by $\Theta = (\lambda_1, \lambda_2, \ldots, \lambda_k)$. During each iteration, GIS constructs a lower bound function to the original log-likelihood function and maximizes it instead.

There exists a particularly simple and analytic solution which solves the auxiliary maximization problem. The parameters obtained from the maximization are guaranteed to improve the original log-likelihood function. There is however one complication for GIS: to ensure that the updates result in monotonic increase in the log-likelihood function, GIS constrains the feature set such that for each event in the training data, $D(x) = C$, where C is a constant and $D(x)$ is defined as the sum of the active features in the event x: $D(x) = \Sigma_{i=1}^{k} f_i(x)$

To satisfy the constraint usually requires the addition of a global correction feature $f_l(x)$, where $l = k + 1$, such that $f_l(x) = C - \Sigma_{i=1}^{k} f_i(x)$. In general, adding new features can affect the model. However, this new correction feature is completely dependent on the other features currently in the feature set. Thus, it adds no new information, and therefore places no new constraints on the model. As a result, the resulting model is unchanged by the addition of the correction feature. However, the rate of convergence of the GIS algorithm is dependent on the magnitude of the constant C: the step size is inversely proportional to the constant C, which implies that the smaller the magnitude of C, the bigger the step size, and the faster the convergence.

Variable Metric Methods (LBFGS)

Malouf (2002) compared the performances of a number of parameter estimation algorithms for the maximum entropy model on a few NLP problems. Malouf (2002) observed that iterative scaling algorithms performed poorly in comparison to first and quasi-second order optimization methods for the NLP problem sets he considered. His conclusion was that a limited memory variable metric algorithm (LBFGS) performed better than the other algorithms on the NLP tasks he considered.

First order methods rely on using the gradient vector $G(\Theta)$ to repeatedly provide estimates of the parameters towards the stationary point at which the gradient is zero and the function value is optimal. Second order optimization techniques, such as Newton's method, improve over first order techniques by using both the gradient and the change in gradient (second order derivatives) when calculating the parameter updates.

The general second-order update rule is calculated from the second-order Taylor series approximation the log-likelihood function, given by:

$L(\Theta + \Delta) \approx L(\Theta) + \Delta^T G(\Theta) + \frac{1}{2}\Delta^T H(\Theta)\Delta$

where $H(\Theta)$ is the matrix containing second order partial derivatives of the log-likelihood function with respect to Θ, or the Hessian matrix. Optimizing the above approximation function results in the update rule:

$\Delta^{k+1} = H^{-1}(\Theta^k)G(\Theta^k)$

Variable-metric methods are a form of quasi-second-order technique, similar to Newton's method, but rather than explicitly calculating the inverse Hessian matrix, at each iteration, variable-metric methods use the gradient to update and approximate the inverse Hessian matrix and achieves improved convergence rate over first-order methods.

Chapter 4

Our Basic Chinese Word Segmenter

In this chapter, we present the basic set of features, and the character normalization technique we employed for our Chinese word segmenter. Also, we describe the segmentation algorithm we used, which is based on dynamic programming. The segmentation algorithm outputs a sequence of admissible tags for a Chinese sentence. This is required since during the testing phase, the maximum entropy classifier treats each character as one distinct test example and assigns it a probability for each possible class without considering its neighboring class tags.

4.1 Chinese Word Segmenter

The Chinese word segmenter we built is similar to the maximum entropy word segmenter we employed in our previous work (Ng and Low, 2004). Our word segmenter uses a maximum entropy framework and is trained on manually segmented sentences. It classifies each Chinese character given the features

derived from its surrounding context. Each character can be assigned one of 4 possible boundary tags: "*b*" for a character that begins a word and is followed by another character, "*m*" for a character that occurs in the middle of a word, "*e*" for a character that ends a word, and "*s*" for a character that occurs as a single-character word. For example, given the following sentence in (i), the tags assigned to the individual characters will be as follows in (ii). (iii) shows the English translation of the example sentence.

(i) 新华社	记者	陈	泰明	
(ii) *b m e*	*b e*	*s*	*b e*	
(iii) Xinhua Agency	reporter	Chen	Taiming	

The basic features of our word segmenter are similar to those used in our previous work (Ng and Low, 2004):

(a) $C_n(n = -2, -1, 0, 1, 2)$
(b) $C_nC_{n+1}(n = -2, -1, 0, 1)$
(c) $C_{-1}C_1$
(d) $Pu(C_0)$
(e) $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$

In the above feature templates, C_i refers to a Chinese character. Templates (a) – (c) refer to a context of five characters (the current character and two characters to its left and right). C_0 denotes the current character, $C_n(C_{-n})$ denotes the character n positions to the right (left) of the current character. For example, given the character sequence "新华社北京", when considering the

character C_0 "社", C_{-2} denotes "新", C_1C_2 denotes "北京", etc. The punctuation feature, $Pu(C_0)$, checks whether C_0 is a punctuation symbol (such as "?", "–", ","). This is useful since certain punctuation symbols such as "," are good delimiters for a word. For the type feature (e), four type classes are defined: numbers belong to class 1, characters denoting dates ("日", "月", "年", the Chinese characters for "day", "month", "year", respectively) belong to class 2, English letters belong to class 3, and other characters belong to class 4. For example, when considering the character "年" in the character sequence "九零年代R", the feature $T(C_{-2})\ldots T(C_2) = 11243$ will be set to 1 ("九" is the Chinese character for "9" and "零" is the Chinese character for "0"). In the Chinese word segmentation problem, these four defined character types tend to have a certain word formation pattern according to the particular word segmentation standard. For example, in segmentation standards such as the Chinese Treebank (CTB) standard, dates have the word formation pattern "number day/month/year" (e.g., "一月"(January), "二十日"(20th) are two separate words).

Besides these basic features, we also made use of character normalization. We note that characters like punctuation symbols and Arabic digits have different character codes in the ASCII, GB, and BIG5 encoding standard, although they mean the same thing. For example, comma "," is represented as the hexadecimal value `0x2c` in ASCII, but as the hexadecimal value `0xa3ac` in GB. In our segmenter, these different character codes are normalized and replaced by the corresponding character code in ASCII. Also, all Arabic digits are replaced by the ASCII digit "0" to denote any digit. Incorporating character normalization enables our segmenter to be more robust against the use of different encodings to represent the same character. In the absence of character nor-

malization, the word segmenter built would be unable to differentiate between the same characters which are represented with different character codes in the training corpus and the test set.

4.2 Segmentation Algorithm

If we were to just assign each character the boundary tag with the highest probability, it is possible that the classifier produces a sequence of invalid tags (e.g. "m" followed by "s"). To eliminate such possibilities, we implemented a dynamic programming algorithm which considers only valid boundary tag sequences given an input string. The probability of a boundary tag assignment $t_1 \ldots t_n$, given a character sequence $C_1 \ldots C_n$, is defined as follows:

$$P(t_1 \ldots t_n | c_1 \ldots c_n) = \prod_{i=1}^{n} P(t_i | h(c_i))$$

where $P(t_i|h(c_i))$ is determined by the maximum entropy classifier, and $c_1 \ldots c_n$ is the input character sequence. The program tags one sentence at a time and works in a dynamic programming fashion. At each character position i, the algorithm considers each next word candidate ending at position i and consisting of K characters in length ($K = 1, \ldots, 20$ in our experiments). (We restrict the length of a word to 20 characters due to performance considerations and due to the fact that Chinese words very rarely exceed such a length.) To extend the boundary tag assignment to the next word W with K characters, the first character of W is assigned boundary tag "b", the last character of W is assigned tag "e", and the intervening characters are assigned tag "m" (if W consists of only one character, then it is assigned the tag "s").

The pseudocode for the segmentation algorithm using dynamic programming follows that of (Russell and Norvig, 2003) and is given as follows:

function segment(sentence)

/* initialize variables */

$n \leftarrow$ length(sentence)

words$\leftarrow$ empty array of length $n + 1$

best $\leftarrow$ array of length $n + 1$, initially 0

$best[0] \leftarrow 1.0$

/* Form and evaluate probability of each candidate word sequence, each word is up to length M. M=20 in our implementation*/

for $i = 1$ to n **do**

  for $j = i$ down to 1 **do**

    word$\leftarrow$ sentence$[j : i]$

    wLen $\leftarrow$ length(word)

    if wLen $> M$ **then**

      break;

    end if

    if P[word] $\times$ best[i - wLen] $>$ best[i] **then**

      best[i] $\leftarrow$ P[word] $\times$ best[i - wLen]

      words[i] $\leftarrow$ word

    end if

  end for

end for

/*get best valid word sequence */

$i \leftarrow n$

while $i > 0$ **do**

push words[i]+" " onto front of sequence

i $\leftarrow$ i - length(words[i])

end while

return sequence

end function

Chapter 5

Handling the OOV problem

A major difficulty faced by a Chinese word segmenter is the presence of out-of-vocabulary (OOV) words. Segmenting a text with many OOV words tends to result in lower accuracy. We address the problem of OOV words in two ways: using an external dictionary containing a list of predefined words, and using additional training corpora of different segmentation standard

5.1 External Dictionary

The easiest way to obtain new words is through word lists, or lexicons, which are readily available on the Internet. The challenge for us therefore is to optimally combine the knowledge from both sources: whenever we are presented with a sequence of characters, we could base our prediction on the output of the original maximum entropy classifier which is trained on word-segmented corpus, or by looking up the word in an external lexicon. When we find a match in the lexicon, it suggests that the character sequence under question is a word in some context. However, in the current sentence in which the character sequence appears, this may or may not be the case. Moreover, the

dictionary words may have been formed according to another segmentation standard. We incorporate knowledge of the external lexicon as additional features in our maximum entropy classifier.

We used an online dictionary from Peking University downloadable from the Internet[2], consisting of about 108,000 words of length one to four characters. If there is some sequence of neighboring characters around C_0 in the sentence that matches a word in this dictionary, then we greedily choose the longest such matching word W in the dictionary. Let t_0 be the boundary tag of C_0 in W, and $C_1(C_{-1})$ be the character immediately following (preceding) C_0 in the sentence. We then add the following features derived from the dictionary:

(f) $C_n t_0 (n = -1, 0, 1)$

For example, consider the sentence "新华社北京...". When processing the current character C_0 "华", we will attempt to match the following candidate sequences "华", "新华", "华社", "新华社", "华社北", "新华社北", and "华社北京" against existing words in our dictionary. Suppose both "华社" and "新华社" are found in the dictionary. Then the longest matching word W chosen is "新华社", t_0 is m, C_{-1} is "新", and C_1 is "社".

5.2 Additional Training Corpora

The presence of different standards in word segmentation limits the amount of training corpora available for the community, due to different organizations

[2]http://ccl.pku.edu.cn/doubtfire/Course/Chinese%20Information%20Processing/Source_Code/Chapter_8/Lexicon_full_2000.zip

preparing training corpora in different segmentation standards. Indeed, if the segmentation standards were the same, there would be no lack of training data, implying that the OOV problem would be significantly reduced. If we could actually incorporate additional training data from other segmentation standards through some methods, we could actually build up a large corpus of training data, and help reduce the OOV problem in Chinese word segmentation.

This extra training data could be thought as a slightly noisy training corpus which contains a certain percentage of corrupted noisy data with wrong segmentation tags assigned for some of the characters. Naively adding all the additional data into the base training set would corrupt the training set with noise, and may reduce the overall predictive accuracy (see Section 6.2.3 for some initial experiments detailing the effect of naively adding additional data of a different segmentation standard). Thus the key problem to using such additional standard set is the need to clean the data set and select only the noise free extra training samples from the additional training data. The method we use to select the relevant extra training data is derived from a technique proposed by (Brodley and Friedl, 1999). Brodley and Friedl (1999) have illustrated that for class noise levels of less than 40%, removing mislabeled instances from the training data can result in higher predictive accuracy relative to classification accuracies achieved without cleaning the training data. Noise elimination is motivated by techniques for removing outliers in regression analysis. Outliers are data instances that do not follow the same model as the rest of the data and appear as though they belong to a different data distribution.

The general procedure makes use of a set of classifiers formed from part of the training data to test whether instances in the remaining part of the

training data are mislabeled and can be briefly described as follows: Assume a noisy training set, with noisy training instances distributed in the training data. Perform n-fold cross validation on the training data. Apply m learning algorithms (known as filter algorithms) to train each train portion of the 10 fold cross validation. Then m resulting classifiers are used to tag each test instance in each testing portion of the respective 10 fold cross validation. If the instance is not tagged correctly, it is considered mislabeled. There are two main variants of the noise elimination procedure. One way would be to use a single algorithm as filter, while the other would be to use an ensemble of filters. In the case of the ensemble filters technique, majority voting or consensus voting can be applied. In majority voting, an instance is classified as mislabeled if a majority of the filters classify the instance as mislabeled. In the case of consensus voting, an instance is considered mislabeled only if all filters classify it as being mislabeled. These mislabeled instances are removed and the final filtered training set is used to train the final classifier. Figure 5.1 shows the general procedure of noise elimination.

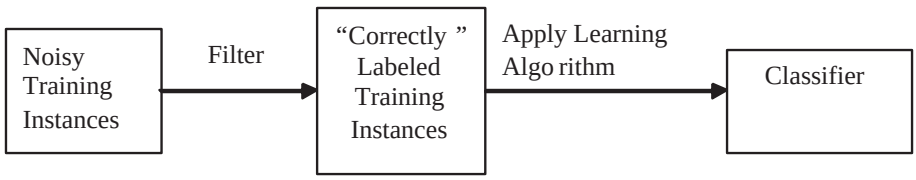

Figure 5.1: General Procedure for noise elimination

We adopt the approach of using the single algorithm filter. The same learning algorithm is used to build both the filter and the final classifier. Our

problem is simplified in that the base set of training data can be assumed to be noise-free (i.e., with negligible errors). Thus we could use the original training data to build our filter (base segmenter), without worrying that our base segmenter is corrupted with noisy samples. The additional training corpus of a different segmentation standard, consisting of noisy samples, is then filtered through our base segmenter to remove the outliers, which we take to be the noisy training samples. Finally, the extra non-noisy training samples and the original training data are combined into a large data set used to train the final classifier.

One practical concern in applying the above technique to obtain extra training data is that the examples selected could be extremely large in number if we are using a large amount of training data gathered from many sources. This could potentially increase the time to train the final classifier. Thus, it would only be sensible if we select the most useful subset from this large extra training set of different segmentation standards, and use it to supplement the existing training corpus. This is based on the concept of active leaning. Active learning acquires labeled data incrementally, using the model learned so far to select the more helpful additional training examples for labeling and training the model. When successful, active learning allows us to reduce the number of training instances required to induce an accurate training model for classification.

The general process of active learning is as follows: We assume that we have a pool L of labeled samples and another pool UL of unlabeled samples. For active leaning, a classifier is first trained on an initial pool L of labeled examples. Next, each candidate sample from the unlabeled pool UL is considered for the labeling process in each phase until some predefined condition is

met. The candidate example is assigned an effectiveness score ES_i, reflecting how useful the sample would be if it is to be incorporated into the training set. Candidate examples above a certain predetermined threshold and deemed most useful are then labeled (e.g. by a human expert) and incorporated into the training set L for subsequent classifier retraining at each phase. Owing to computational constraints, usually a set of candidate samples (instead of only one candidate sample) is considered during each phase, and a limit of y most useful samples may be selected during each phase for retraining purposes.

For efficiency reasons, in our implementation, we select all the new training samples with assigned probability (by the maximum entropy classifier) below a certain probability threshold in one single step, instead of incremental selection with retraining at each phase. Extra training samples predicted with a high confidence are considered to be very similar to the original training samples, and therefore less useful to be incorporated since the original training data set already has very similar training samples. Also, no relabeling by human experts is done. We just assume that the additional selected training examples are correctly labeled and all noisy data has been filtered during the noise elimination process. Thus the entire selection process is completely automatic, with no need for human intervention or additional manual work.

The main steps in our proposed scheme in selecting the extra training data are depicted in Figure 5.2. Specifically, the steps taken are:

1. Perform training with maximum entropy modeling using the original training corpus D_0 annotated in a given segmentation standard.

2. Use the trained word segmenter to segment another corpus D_i annotated in a different segmentation standard.

3. Suppose a Chinese character C in D_i is assigned a boundary tag t by

the word segmenter with probability p. If t is identical to the boundary tag of C in the gold-standard annotated corpus D_i, and p is less than some threshold θ, then C (with its surrounding context in D_i) is used as additional training data.

4. Add all such characters C as additional training data to the original training corpus D_0, and train a new word segmenter using the enlarged training data.

5. Evaluate the accuracy of the new word segmenter on the same test data annotated in the original segmentation standard of D_0.

For the tests on bakeoff 2 data, when training a word segmenter on a particular training corpus, the additional training corpora are all the three corpora in the other segmentation standards. For example, when training a word segmenter for the AS corpus, the additional training corpora are CITYU, MSR, and PKU. Similarly for our tests on bakeoff 1 data, when training a word segmenter on a particular training corpus, the additional training corpora are all the three corpora in the other segmentation standards present in the bakeoff 1 data set. The necessary character encoding conversion between GB and BIG5 is performed, and the probability threshold θ is set to 0.8 for our final segmenter. In Section 6.2.4, we will present empirical results indicating that setting θ to a higher value does not further improve segmentation accuracy, but would instead increase the training set size and incur longer training time.

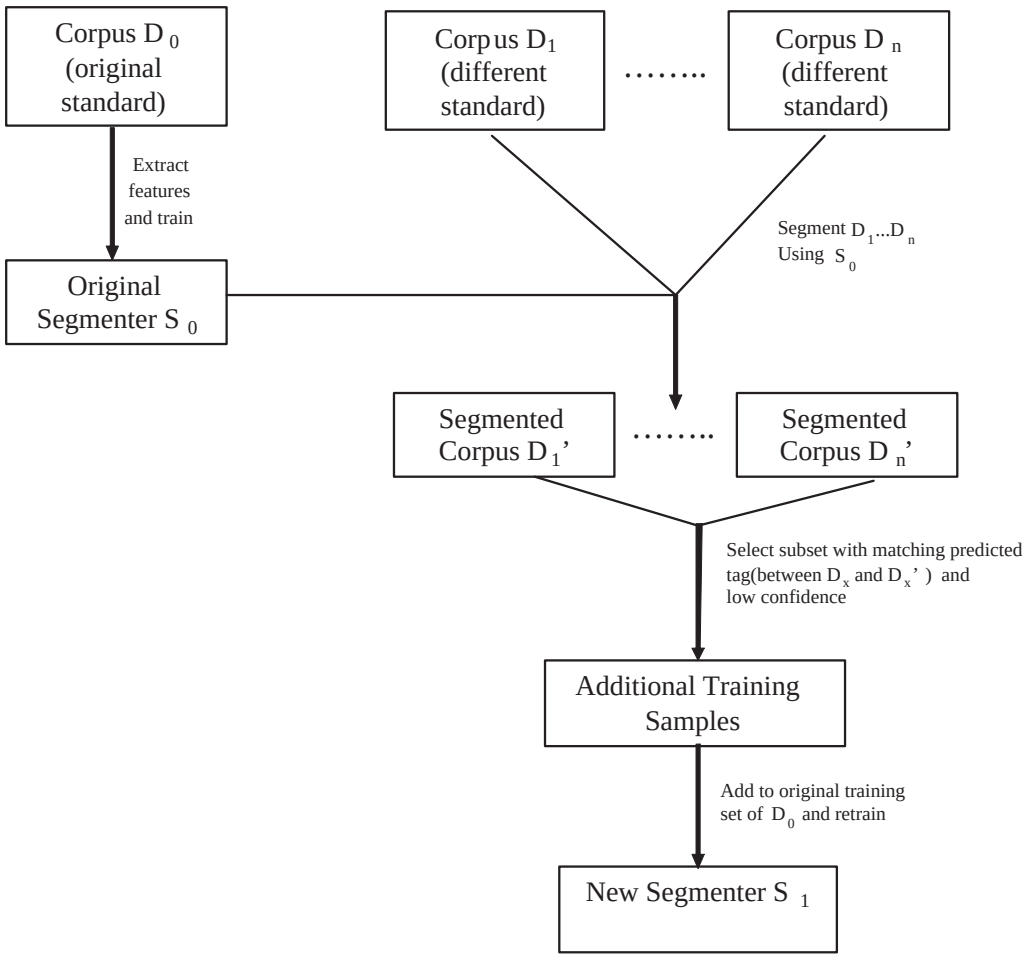

Figure 5.2: Selection of extra data for retraining

Chapter 6

Experiments on SIGHAN Datasets

In this chapter, we present the results of experiments we conducted using the 8 datasets from the First and Second International Chinese Word Segmentation Bakeoff. The experiments we conducted include using the basic features presented in Section 4.1, the basic+dict features presented in Section 5.1, and evaluating the effect of adding noise-filtered additional training corpora to supplement the original training data (example selection) presented in Section 5.2.

6.1 SIGHAN Chinese Word Segmentation Bakeoff

Prior to the organization of SIGHAN's First International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003), comparison of different approaches to Chinese word segmentation across systems was difficult due to the lack of standardized test sets. Many word segmentation standards exist,

including five different segmentation standards (Academia Sinica (AS), Hong Kong City University (CITYU), UPenn Chinese Treebank (CTB), Microsoft Research (MSR), and Peking University (PKU)) that were utilized in the two bakeoffs. Since many papers were based on their own training and test sets, it was hard to draw a conclusion as to which method was truly superior and also if it would perform equally well on another corpus of a different segmentation standard. In order to enable a clear comparison between our segmenter and the others presented in other recent Chinese word segmentation research, the experiments we conducted for our Chinese word segmenter are all based on the datasets obtained from the First and Second International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003; Emerson, 2005).

The first SIGHAN bakeoff provided corpora of four different standards, detailed in Table 6.1. The second SIGHAN bakeoff provided another new corpus from MSR, together with 3 of the standards already used in bakeoff 1. Details of the bakeoff 2 corpora are provided in Table 6.2. The SIGHAN bakeoff allowed participants to participate in the open or closed track. In the open track, participants could use external knowledge sources to supplement the training corpus, while the closed track allowed participants to use only the individual training corpus to train their segmenter.

Corpus	Encoding	#Train Words	#Test Words	Test OOV
AS	Big 5	5.8M	12K	0.022
CITYU	Big 5	240K	35K	0.071
CTB	EUC-CN	250K (GB 2312-80)	40K	0.181
PKU	GBK	1.1M	17K	0.069

Table 6.1: SIGHAN Bakeoff 1 Data

Corpus	Encoding	#Train Words	#Test Words	Test OOV
AS	Big 5 Plus	5.45M	122K	0.043
CITYU	BIG 5/HKSCS	1.46M	41K	0.074
MSR	CP936	2.37M	107K	0.026
PKU	CP936	1.1M	104K	0.058

Table 6.2: SIGHAN Bakeoff 2 Data

Results from the participants of the SIGHAN bakeoff 1 indicated that no one participant performed consistently better than all others. From the results of the closed category, it was noted that out-of-vocabulary (OOV) words had a significant impact on the accuracy. The CTB closed track, with the test corpus containing an OOV of 18.1% reported the lowest accuracy in general, with the best system reporting an accuracy of 88.1%. On the other hand, the AS corpus, with a OOV of only 2.2%, had a high accuracy of 96.1% from the top team.

6.2 Experimental Results

We carried out our experiments on the SIGHAN bakeoff 1 and 2 training and test sets. We evaluated our segmenter on all the 4 corpora for bakeoff 1: Academia Sinica (AS), City University of Hong Kong (CITYU), Chinese Treebank (CTB), and Peking University (PKU) for the open category. We repeated the experiments for all the 4 corpora in bakeoff 2: AS, CITYU, Microsoft Research (MSR), and PKU. The Java-based opennlp maximum entropy package v2.1.0 from sourceforge[3] was employed as the GIS version, while another C++ Maximum Entropy package (v20041229) from Le Zhang of Edinburgh Univer-

[3]http://maxent.sourceforge.net/

sity [4] was employed as the LBFGS version. Training was done with a feature cutoff of 2 (except for the AS corpus in bakeoff 1 and 2, in which we applied cutoff 3) and 100 iterations for the GIS version, while Gaussian prior variance of 2.5 and 1000 iterations were selected for the LBFGS version. The usual three measures: recall, precision, and F-measure are used to evaluate the accuracy of our word segmenter. To define the three measures, we use the follow definitions:

N	Number of words occurring in the gold hand-segmented text
c	Number of words correctly identified by the word segmenter
n	Number of words identified by the word segmenter

The measures: recall, precision, and F-measure are defined as:

$$recall = \frac{c}{N}$$

$$precision = \frac{c}{n}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$

The above word segmentation recall (R), precision (P), and F-measure are then measured using the official scorer used in the SIGHAN bakeoff (Sproat and Emerson, 2003; Emerson, 2005).

For all the tabulated results in the following tables, Version V1 used only the basic features (Section 4.1); Version V2 used the basic features and additional features derived from our external dictionary (Section 5.1); Version V3 used the basic features plus additional training corpora (Section 5.2); and Version V4 is the version combining basic features, external dictionary, and additional training corpora.

[4] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

6.2.1 Basic Features and Use of External Dictionary

We carried out a series of experiments using bakeoff 1 and 2 data to test the effectiveness of our word segmenter. Table 6.3 and Table 6.4 give the results of word segmentation using the basic features described in Section 4.1 and dictionary features described in Section 5.1 for bakeoff 1 and 2 respectively.

Corpus	GIS V1	GIS V2	LBFGS V1	LBFGS V2
AS	0.967	0.968	0.969	0.970
CITYU	0.940	0.959	0.945	0.960
CTB	0.861	0.893	0.869	0.900
PKU	0.954	0.967	0.953	0.967

Table 6.3: V1 and V2 bakeoff 1 word segmentation accuracy (F-measure) for GIS and LBFGS parameter estimation algorithm

Corpus	GIS V1	GIS V2	LBFGS V1	LBFGS V2
AS	0.950	0.953	0.954	0.955
CITYU	0.948	0.958	0.954	0.962
MSR	0.960	0.969	0.965	0.972
PKU	0.948	0.966	0.950	0.967

Table 6.4: V1 and V2 bakeoff 2 word segmentation accuracy (F-measure) for GIS and LBFGS parameter estimation algorithm

While the training time and the number of iterations required for LBFGS parameter estimation algorithm is more than those for GIS, overall accuracy indicates that LBFGS is a slightly better parameter estimation algorithm for the Chinese word segmentation task. For all the above runs, LBFGS parameter estimation algorithm has obtained a higher F-measure than GIS on the same test sets. Also, the use of dictionary consistently improves segmentation

accuracy.

6.2.2 Usefulness of the Additional Training Corpora

Additional training corpora of different segmentation standards can provide useful training samples and context features to supplement the original training corpus, but in order for them to be useful, there must be some form of similarity in segmentation standards for both corpora, so that useful samples can be selected from the additional training corpus. Although different segmentation standards exist in Chinese word segmentation, we note that many words will still be segmented in the same way. For example, consider the word "愉快"(happy), the meaning of the word would be lost if this word was separated into two words, so the different segmentation standards will still segment such words in the same way.

As a gauge to estimate the usefulness of training corpora of different segmentation standards, we carried out the following procedure:

1. Perform training with maximum entropy modeling using a particular training corpus A.

2. Use the trained word segmenter to segment the other 3 testing data sets B, C, D (of different segmentation standards) for the respective bakeoff.

3. Measure the accuracy of the segmented test data sets B, C, D, against their corresponding gold standard annotation.

The accuracy of the segmented test data provides a gauge of the usefulness of training corpora of different segmentation standards. Table 6.5 and Table 6.6 show the results of our experiments on bakeoff 1 and bakeoff 2 data. To enable quicker experiments with shorter training time, experiments were

conducted using basic features and GIS parameter estimation algorithm for ME modeling. In Table 6.5 for example, table entry with row AS and column CTB refers to the F-measure obtained on CTB test set by using a segmenter trained with AS training set. The results indicate that even if a corpus of a different segmentation standard is used to train the segmenter, over 80% in F-measure can still be obtained. Thus we can see that the additional training corpora contain useful information that can aid in word segmentation.

Train Corpus	AS	CITYU	CTB	PKU
AS	0.967	0.889	0.912	0.856
CITYU	0.874	0.940	0.846	0.822
CTB	0.866	0.848	0.861	0.834
PKU	0.877	0.862	0.847	0.954

Table 6.5: Word segmentation accuracy (F-measure) on bakeoff 1 test data obtained using training data of a different segmentation standard

Train Corpus	AS	CITYU	MSR	PKU
AS	0.950	0.884	0.829	0.877
CITYU	0.892	0.948	0.831	0.881
MSR	0.831	0.811	0.960	0.851
PKU	0.847	0.856	0.859	0.948

Table 6.6: Word segmentation accuracy (F-measure) on bakeoff test 2 data obtained using training data of a different segmentation standard

6.2.3 Naive Use of Additional Training Corpora

Based on the tests carried out in 6.2.2, we can see that corpora of different segmentation standard can still provide useful information in word segmentation.

As a first try to test the effect of just adding additional training corpora of different segmentation standard to supplement the original training data, we first implemented a naive retraining scheme. In this naive retraining scheme, we just added all the training corpora of the other segmentation standards to the original training corpus and tested the performance of the training on the whole training set, using the ME approach with the basic feature set. For this set of experiments, we just conducted it using GIS parameter estimation algorithm to enable quicker experiments. Results are shown in Table 6.7 for bakeoff 1 data and Table 6.8 for bakeoff 2 data. In Table 6.7 for example, table entry AS+CTB refers to the F-measure obtained on AS test set by using the segmenter trained with original AS training set and supplemented with CTB training corpus. Table entry AS+AS refers to the F-measure obtained on AS test set using segmenter trained with the original AS training data. As shown from the results, except for CTB (which does benefit from using additional training corpus), such a naive approach usually results in a drop in F-measure for the other 3 corpora. Naively adding training data from different standards ultimately results in too much noise due to the incorporation of wrongly segmented words in training data as a consequence of different word segmentation standards and results in a drop in accuracy. This demonstrates the necessity of filtering out the noisy data of the additional training corpora using the noise elimination method we introduced in Section 5.2.

6.2.4 Usefulness of Example Selection

As part of our experiments to determine the usefulness of example selection, we carried out experiments using bakeoff 1 and 2 data with different thresholds to determine the usefulness of selecting additional training corpora of different

Corpus	+AS	+CITYU	+CTB	+PKU
AS	0.967	0.968	0.967	0.965
CITYU	0.919	0.940	0.933	0.921
CTB	0.919	0.878	0.861	0.862
PKU	0.936	0.951	0.949	0.954

Table 6.7: Word segmentation accuracy (F-measure) for bakeoff 1 data obtained from adding additional training data from another corpus of a different segmentation standard, with the GIS parameter estimation algorithm. Note that the original results without retraining are obtained from the center diagonal (AS+AS for example)

Corpus	+AS	+CITYU	+MSR	+PKU
AS	0.950	0.949	0.937	0.948
CITYU	0.932	0.948	0.892	0.935
MSR	0.930	0.946	0.960	0.937
PKU	0.928	0.934	0.883	0.948

Table 6.8: Word segmentation accuracy (F-measure) for bakeoff 2 data obtained from adding additional training data from another corpus of a different segmentation standard, with the GIS parameter estimation algorithm

segmentation standards when applied to both the basic and basic+dict set of features. Table 6.9 and Table 6.10 (for version V3) detail the results of word segmentation with example selection at different thresholds with basic features for bakeoff 1 and 2 respectively, while Table 6.11 and Table 6.12 give the accuracy of word segmentation with example selection at different thresholds using the basic+dict features for bakeoff 1 and 2 respectively.

Our results indicate that using a higher probability threshold than 0.8 does not really help in improving accuracy, but instead just incur extra training time and memory. Thus with a large supply of additional data, it would not

be realistic to just use all the extra training data procured. Instead using a threshold of 0.8 suffices.

Also our proposed method to make use of example selection works best for a small corpus like CTB with a small training data set and high OOV test set. The additional training data incorporated helps to achieve a significant increase in accuracy.

Corpus	0.5	0.6	0.7	0.8	0.9
AS	0.969	0.970	0.969	0.970	0.969
CITYU	0.954	0.955	0.955	0.955	0.954
CTB	0.913	0.915	0.918	0.917	0.915
PKU	0.957	0.957	0.958	0.958	0.957

Table 6.9: Bakeoff 1 V3 word segmentation accuracy (F-measure) at different threshold settings for LBFGS parameter estimation algorithm

Corpus	0.5	0.6	0.7	0.8	0.9
AS	0.954	0.954	0.956	0.956	0.956
CITYU	0.960	0.961	0.961	0.961	0.961
MSR	0.965	0.965	0.965	0.965	0.965
PKU	0.954	0.954	0.955	0.956	0.956

Table 6.10: Bakeoff 2 V3 word segmentation accuracy (F-measure) at different threshold settings for LBFGS parameter estimation algorithm

6.2.5 Overall Summary of our Word Segmenter Results

Finally, we present the overall summary performance of our various implementations with bakeoff 1 and 2 training and test datasets using LBFGS parameter estimation algorithm for ME modeling. Tables 6.13 and 6.14 show the sum-

Corpus	0.5	0.6	0.7	0.8	0.9
AS	0.971	0.971	0.971	0.971	0.970
CITYU	0.963	0.963	0.964	0.963	0.964
CTB	0.923	0.923	0.925	0.924	0.924
PKU	0.968	0.968	0.969	0.969	0.970

Table 6.11: Bakeoff 1 V4 word segmentation accuracy (F-measure) at different threshold settings for LBFGS parameter estimation algorithm

Corpus	0.5	0.6	0.7	0.8	0.9
AS	0.956	0.956	0.956	0.956	0.956
CITYU	0.963	0.963	0.964	0.964	0.964
MSR	0.971	0.971	0.971	0.971	0.971
PKU	0.968	0.969	0.969	0.969	0.969

Table 6.12: Bakeoff 2 V4 word segmentation accuracy (F-measure) at different threshold setting for LBFGS parameter estimation algorithm

mary of our results for the different feature implementations we tested on. Also for bakeoff 1, we show the open category results of 2 other systems (Gao *et al.*, 2004; Peng *et al.*, 2004), in which we also perform better than in terms of F-measure. Table 6.15 and 6.16 show the detailed V4 results for bakeoff 1 and 2 respectively. Finally, Figures 6.1 and 6.2 show our V4 LBFGS segmentation accuracy results when compared with other participants of bakeoff 1 and bakeoff 2 respectively. Due to space constraint, the accuracy figures for bakeoff 2 detailed in Figure 6.2 only shows participants who obtained above the baseline accuracy using maximal matching. In our official participation results in bakeoff 2, our word segmenter achieved the highest F-measure for

AS, CITYU, and PKU and the second highest for MSR. Our official bakeoff 2 results are not included in the below figures. Our V4 LBFGS F-measures are either the same or better than our word segmenter's F-measure in the SIGHAN bakeoff 2 participation.

Corpus	LBFGS V1	LBFGS V2	LBFGS V3	LBFGS V4	Best SIGHAN	Gao *et al.* (2004)	Peng *et al.* (2004)
AS	0.969	0.970	0.970	0.971	0.961	0.958	0.957
CITYU	0.945	0.960	0.955	0.963	0.956	0.954	0.946
CTB	0.869	0.900	0.917	0.924	0.912	0.904	0.894
PKU	0.953	0.967	0.958	0.969	0.959	0.955	0.946

Table 6.13: Summary of bakeoff 1 word segmentation accuracy (F-measure) for LBFGS parameter estimation algorithm. Note that the 0.961 for AS is for closed category since the open category achieved a lower F-measure than the closed category in the official bakeoff 1 results

Corpus	LBFGS V1	LBFGS V2	LBFGS V3	LBFGS V4	Best SIGHAN
AS	0.954	0.955	0.956	0.956	0.956(Ours)
CITYU	0.954	0.962	0.961	0.964	0.962(Ours)
MSR	0.965	0.972	0.965	0.971	0.972
PKU	0.950	0.967	0.956	0.969	0.969(Ours)

Table 6.14: Summary of bakeoff 2 word segmentation accuracy (F-measure) for LBFGS parameter estimation algorithm

Corpus	R	P	F	R_{OOV}	R_{IV}
AS	0.971	0.970	0.971	0.744	0.976
CITYU	0.966	0.960	0.963	0.850	0.975
CTB	0.924	0.923	0.924	0.812	0.949
PKU	0.971	0.968	0.969	0.846	0.980

Table 6.15: Our final V4 detailed bakeoff 1 F-measure results

Corpus	R	P	F	R_{OOV}	R_{IV}
AS	0.962	0.951	0.956	0.694	0.974
CITYU	0.967	0.960	0.964	0.840	0.977
MSR	0.971	0.970	0.971	0.752	0.977
PKU	0.967	0.970	0.969	0.846	0.975

Table 6.16: Our final V4 detailed bakeoff 2 F-measure results

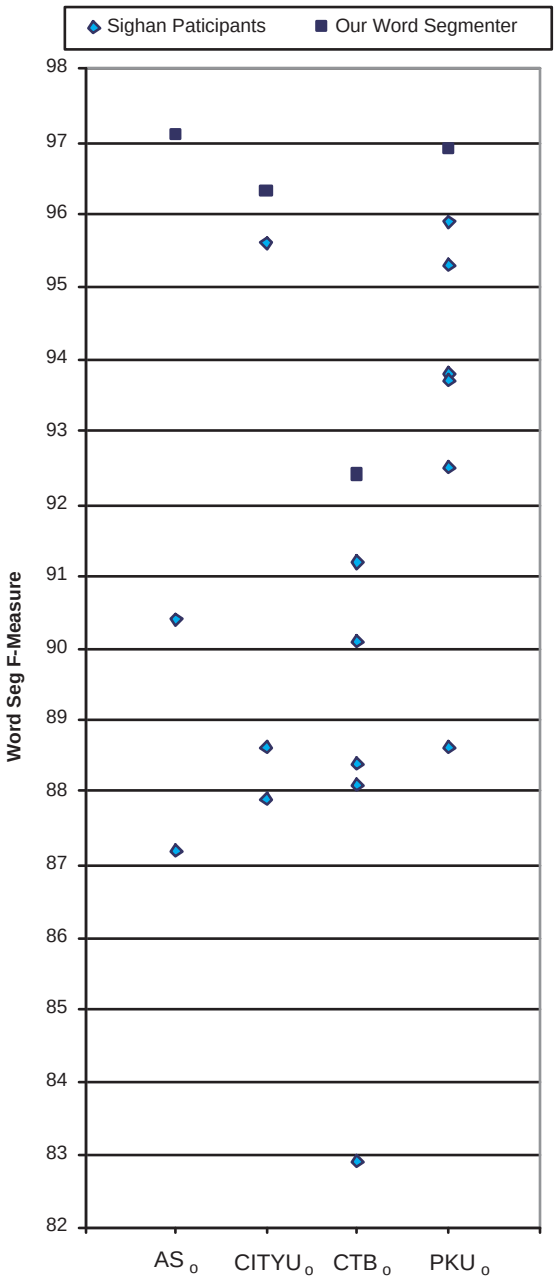

Figure 6.1: Our final V4 word segmenter F-measure when compared with other bakeoff 1 participants in the open category. Note that the highest F-measure obtained for AS was in closed category at 0.961, but still lower than our best result

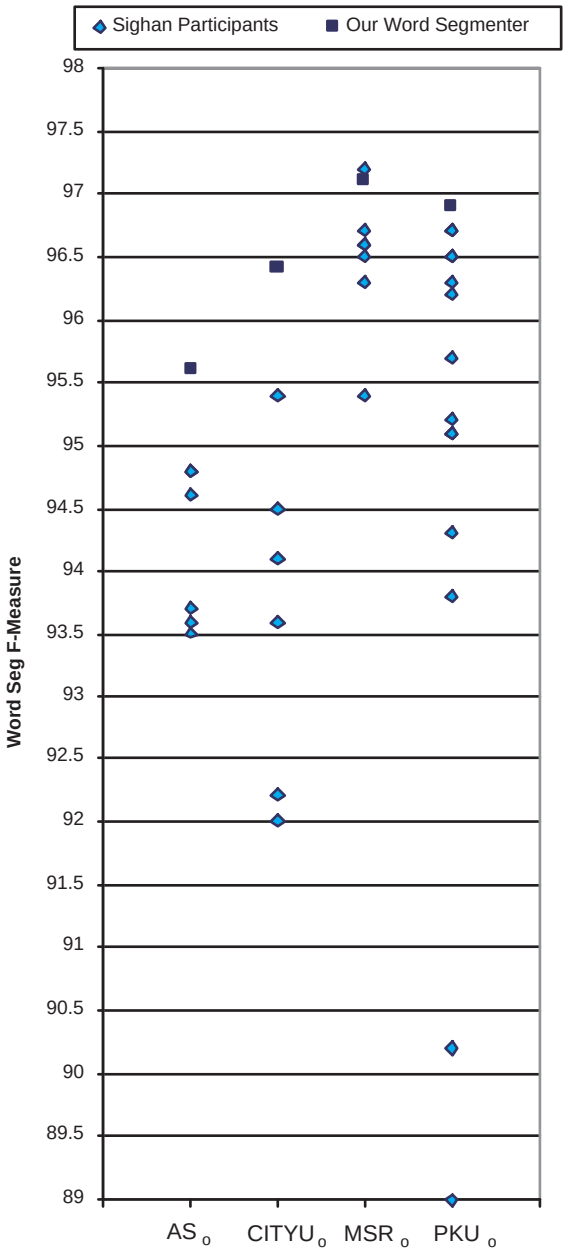

Figure 6.2: Our final V4 word segmenter F-measure when compared with other bakeoff 2 participants in the open category

Chapter 7

Discussions and Conclusions

7.1 Conclusions

Using a maximum entropy approach, our Chinese word segmenter achieves state-of-the-art accuracy, when evaluated on all the corpora in the open track of the First and Second International Chinese Word Segmentation Bakeoff. In the Open category of the Second International Chinese Word Segmentation Bakeoff in which we officially participated in, our word segmenter's accuracy ranked top in three corpora (AS, CITYU, and PKU), and second in one corpus (MSR). In order to handle the OOV problem, we managed to come up with two general methods to handle OOV words. The methods we introduced are general enough to work for all the test corpora we tested on, yet simple to implement.

An external dictionary is used to add three simple features, $C_n t_0 (n = -1, 0, 1)$ to the original set of features. These features are not designed based on or tuned to any segmentation standard. Overall, it works well for all the different corpora we tested.

We also used additional training corpora of different segmentation stan-

dards to supplement the original given training data set. By a process of noise filtering and active sampling, we are able to obtain useful extra training data to supplement the original training data. Corpora of different segmentation standards are readily available, and by using our proposed method, we can effectively pool many different knowledge resources for the word segmentation task. From our experiments, this method is shown to work especially well for the CTB corpus, a small training data set with an observed high OOV in the test set.

7.2 Recommendations for Future Work

A further investigation of the effectiveness of different supervised learning approaches for the Chinese word segmentation task could be performed. In this thesis, we only compared the differences in performance between GIS and LBFGS parameter estimation algorithms within the maximum entropy modeling framework. Within Chinese word segmentation, we have researchers adopting different learning algorithms such as Conditional Random Fields(CRFs) (Tseng *et al.*, 2005), Perceptron Learning (Li *et al.*, 2005) for the same task. A more conclusive comparison of the different supervised learning approaches for the Chinese word segmentation task could be conducted as an extension to the work we presented.

Our proposed use of additional training corpora to supplement existing training data for the Chinese word segmentation task has been shown to work generally well for all the experiments we performed on the Chinese word segmentation task. Another possible area we could work on would be to extend this method of acquiring additional training data to other tasks such as Part-of-speech (POS) tagging and Named Entity Recognition (NER) for Chinese.

However, due to different POS tags and NER tags used in different resources, there is a need to try to unify them in some way for this method to work. The data available for the above mentioned tasks is significantly lesser than what is available for the Chinese word segmentation task, thus the usefulness of acquiring extra data may be even greater if we can successfully extend it to these tasks.

Bibliography

Masayuki Asahara, Kenta Fukuoka, Ai Azuma, Chooi-Ling Goh, Yotaro Watanabe, Yuji Matsumoto, and Takashi Tsuzuki. Combination of machine learning methods for optimum chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 134–137, 2005.

Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

John Broglio, Jamie P. Callan, and W. Bruce Croft. Technical issues in building an information system for chinese. Ciir technical report ir-86, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, 1996.

Keh-Jiann Chen and Shing-Huan Liu. Word identification for mandarin chinese sentences. In *Proceedings of 14th International Conference on Computational Linguistics(COLING 1992)*, pages 101–107, 1992.

Kwok-Shing Cheng, Gilbert H. Young, and Kam-Fai Wong. A study on word-based and integral-bit chinese text compression algorithms. *Journal of the American Society for Information Science*, 50(3):218–228, 2003.

Hai Leong Chieu and Hwee Tou Ng. Named entity recognition: A maximum

entropy approach using global information. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 190–196, 2002.

Yubin Dai, Christopher S. G. Khoo, and Teck Ee Loh. A new statistical formula for chinese text segmentation incorporating contextual information. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 82–89, 1999.

J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

Thomas Emerson. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 123–133, 2005.

F. Erik, Kim Sang Tjong, and Buchholz Sabine. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132, 2000.

Jianfeng Gao, Andi Wu, Mu Li, Chang-Ning Huang, Hongqiao Li, Xinsong Xia, and Haowei Qin. Adaptive chinese word segmentation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics(ACL 2004)*, 2004.

Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. Chinese word segmentation by classification of characters. In *Proceedings of the Third SIGHAN Workshop*, 2004.

Yaoyong Li, Chuanjiang Miao, Kalina Bontcheva, and Hamish Cunningham. Perceptron learning for chinese word segmentation. In *Proceedings of the*

Fourth SIGHAN Workshop on Chinese Language Processing, pages 154–157, 2005.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. A maximum entropy approach to chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 161–164, 2005.

Xiaoqiang Luo. A maximum entropy chinese character-based parser. In *Proceedings of the 2003 Conference on Emprical Methods in Natural Language Processing*, pages 192–199, 2003.

Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55, 2002.

Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 277–284, 2004.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING 2004)*, 2004.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging.

In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 1996.

S.J. Russell and P. Norvig. *Artifical Intelligence: A Modern Approach*. Prentice Hall, second edition, 2003.

Richard Sproat and Thomas Emerson. The first international chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143, 2003.

Richard Sproat and Chilin Shih. A statistical method for finding word boundaries in chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351, 1990.

Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. A stochastic finite-state word segmentation algorithm for chinese. *Computational Linguistics*, 22(3):377–404, 1997.

W.J. Teahan, Yingying Wen, Rodger J. McNab, and Ian H. Witten. A compression-based algorithm for chinese word segmentation. *Computational Linguistics*, 26(3):375–393, 2000.

Huihsin Tseng, Pichuan Chang, Galen Andrew, and Manning Christopher Jurafsky, Daniel. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168–171, 2005.

Hanna Wallach. Efficient training of conditional random fields. Master's thesis, Division of Informatics, University of Edinburgh, Edinburgh, U.K., 2002.

Nianwen Xue and Libin Shen. Chinese word segmentation as LMR tagging.

In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 176–179, 2003.