**Project Report**

# Exploiting Additional Training Corpora for Chinese Word Segmentation

**Lai Xiao Ni**

School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543

## Abstract

One approach to obtain an improved Chinese Word segmenter is to incorporate additional training corpora of other segmentation standards into the original training corpus, and conduct retraining of the model. This project implements this approach on a Chinese word segmenter within maximum entropy framework and proves that such an approach is effective under certain conditions only.

## 1 Introduction

This project is part of the CS2306S module requirement. It is in relation to the SIG group, Natural Language Processing, in Special Program in Computing (SPC). The project mainly replicates the experiment of building an improved version of Chinese word segmenter within its maximum entropy framework which successfully reduces the problem of OOV words (Low and Ng, 2006), with focus on the method of using additional training corpora. The project also explores how to eliminate noise in training corpora in order to more effectively increase the accuracy of Chinese word segmentation, and proposes possible explanation for this issue.

## 2 Background

### 2.1 Motivation

Unlike English text which consists of words with unambiguous boundaries of white spaces in between, Chinese text is a string of un-segmented Chinese characters without clearly-marked word boundaries. It is recognized that a Chinese character can exist as a single word itself or as part of a multi-character word. Such a characteristic of Chinese language makes word segmentation a necessary first step before embarking on more complex Natural Language Processing (NLP) tasks such as machine translation and text-to-speech synthesis.

However, Chinese word segmentation is never an easy task. One major difficulty it faces is the out-of-vocabulary (OOV) words. This problem arises inevitably due to appearance of new Chinese words constantly. Although many previous works had

been done by researchers to build an accurate Chinese word segmenter, there is still room for further improvement if the issue of OOV words is taken into consideration during the design of a Chinese word segmenter.

### 2.2 Previous Work

In the past two decades, many research works have been done on Chinese word segmentation. Some effective methods have been invented: Dictionary-Based approach (Chen and Liu, 1992), which basically uses a fixed word list and some grammar rules; and Simple Statistical approach (Sproat and Shih, 1990), which groups characters with the highest mutual information. Both approaches have the advantage of simplicity and efficiency. Later a hybrid method, which used a statistical approach with the guidance of a dictionary, was introduced to improve the segmentation accuracy.

Subsequently, more successful methods, which involved some supervised machine learning approaches (Luo, 2003; Ng and Low, 2004; Peng et al., 2004; Xue and Shen, 2003) have been studied. These methods include: the Conditional Random Fields (CRFs) method applied by Peng et al. (2004) which included a separate OOV detection phase to detect OOV words in the test data; the Support Vector Machines (SVMs) used by Goh et al. (2004) which produced an SVM classifier that can reassign the word boundaries based on the results of forward maximal matching (FMM) algorithm and backward maximal matching (BMM) algorithm; and the Transformation-Based Learning (TBL) method which used a TBL postprocessor that can transform segmenter to adapt to a new segmentation standard.

Besides the above three, there is one more important machine learning approach —the Maximum Entropy (ME) approach. This is the one that would be used during the implementation of this project. In this approach, each Chinese character is denoted with one of the following four tags to indicate its relative position in a word:

| tag | meaning | example |
|---|---|---|
| b | Character that begins a word | “超” in “超市” |
| m | Character in the middle of a word | “华” in “新华社” |
| e | Character that ends a word | “纪” in “世纪” |
| s | Character that occurs as a single-character word | “我” |

In fact, the process of Chinese word segmentation is basically a process of determining the highest conditional probability p(y|x) which predicts the chance of class y occurring based on history or context x. But it is impossible to have a training corpus so large such that all possible (x, y) pairs are included. This results in many difficulties, such as the presence of a large number of pairs with zero occurring probability. Thus the ME approach comes into play. The Principle of Maximum Entropy states that when one has only partial information about the possible outcomes, one should choose the probabilities so as to maximize the uncertainty

about the missing information, as shown by Jaynes. In this case, the probability distribution function has its entropy maximized and manifests itself as the following form (Pietra et al., 1997):

$$P(y \mid x) = \frac{1}{Z(x)} \prod_{i=1}^{k} \lambda_i^{f_i(x,y)}$$

where y is the outcome class, x is the history observed, Z(x) is a normalization constant, $f_i(x,y) \in \{0,1\}$ is a feature function, and $\lambda_i$ is a weight corresponding to feature $f_i$.

Two most commonly used algorithms estimating the parameters $\lambda_i$ are Generalized Iterative Scaling (GIS) and a limited memory variable metric algorithm (LGFGS). In this project, the latter algorithm was adopted as the parameter estimation algorithm because it was found to perform better during the Chinese word segmentation task.

# 3 My Approach

## 3.1 Data

The first and second SIGHAN Bakeoff data are the data with which this project begins. They were employed as word segmentation standards during the First and Second International Chinese Word Segmentation Bakeoff organized by SIGHAN in 2003 and 2005 respectively (Sproat and Emerson, 2003; Emerson, 2005). Before they were introduced, there was no standardized test sets in the field of Chinese word segmentation; and researchers from different organizations found it hard to compare performances of their word segmenters, including how efficient or effective the segmentation method can work and how well it can perform even on corpus of a different segmentation standard.

Bakeoff 1 consists of four different word segmentation standards: Academia Sinica (AS), Hong Kong City University (CITYU), Penn Chinese Treebank (CTB), and Peking University (PKU). Bakeoff 2 also consists of four segmentation standards, replacing CTB with a new corpus from Microsoft Research (MSR). Each of the word segmentation standards provides a training corpus and a test set. Details of these standards are as follows (Ng and Low, 2006). Test OOV refers to the percentage of OOV words in the test set.

| Corpus | Encoding | #Train Words | #Test Words | Test OOV |
|---|---|---|---|---|
| AS | Big5 | 5.8M | 12K | 0.022 |
| CITYU | Big5 | 240K | 35K | 0.071 |
| CTB | GB | 250K | 40K | 0.181 |
| PKU | GBK | 1.1M | 17K | 0.069 |

**Figure 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 | 1.45M | 41K | 0.074 |
| MSR | CP936 | 2.37M | 107K | 0.026 |
| PKU | CP936 | 1.1M | 104K | 0.058 |

**Figure 2**
SIGHAN bakeoff 2 data

The size and Test OOV percentage of different standards are later found to be important factors in determining how effective incorporating corpora of different segmentation standards to original segmenter can improve the segmentation accuracy.

## 3.2 Tools Used Throughout Experiment

### 3.2.1 Big5GB Converter

This is a freeware downloadable from internet which can convert plain text from GB coding to Big5 coding. As what the tables above indicate, the corpora of different standards resemble in the form of either GB coding or Big5 coding (CP936 can be seen as GBK encoding working on the Win2K system). However, the encoding format is not important in this experiment because our task is to segment Chinese characters, regardless of how they are represented in computers.

### 3.2.2 Tembusu

Tembusu is a Linux-based Compute Cluster catering to parallel and distributed computing research, etc. built by School of Computing, NUS. The whole experiment was carried out on the Tembusu server.

## 3.3 Original Chinese Word Segmenter

The Chinese word segmenter is built using a maximum entropy approach. This is achieved by following a few steps.

Firstly, basic features from manually segmented Chinese text, which can be the training corpus of one particular segmentation standard, are extracted. The basic features used in this project follow the ones used in the work of Ng and Low (2004):

a) $C_n$ (n = -2, -1, 0, 1, 2)
b) $C_n\ C_{n+1}$ (n = -2, -1, 0, 1)
c) $C_{-1}\ C_1$
d) $P_u(C0)$
e) $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$

in which $C_i$ refers to the Chinese character with relative position i with respect to the current character, $P_u$ refers to the punctuation of the specified character and T represents the type of the specified character. There are four types of T: class 1, numbers; class 2, dates-denoting characters; class 3, English letters; and class 4, other characters.

Extra dictionary is used to improve the segmentation accuracy. If dictionary is used, one additional feature would be created:

(f) $C_nT_0$ (n = -1, 0, 1)

This is used to improve segmentation accuracy. But its working will not be discussed in this experiment. Extra dictionary is used in all segmenter trainings for consistency purpose.

Character normalization is also carried out to resolve the problem caused by different encoding techniques.

```
b a=上 b=_ c=_ d=海 e=浦 f=_上 g=上海 h=__ i=海浦 j=_海 p=44444 r=__b s=上_b t=海_b
e a=海 b=_ c=上 d=浦 e=东 f=上海 g=海浦 h=_上 i=浦东 j=上浦 p=44444 r=上_e s=海_e t=浦_e
b a=浦 b=上 c=海 d=东 e=开 f=海浦 g=浦东 h=上海 i=东开 j=海东 p=44444 r=海_b s=浦_b t=东_b
e a=东 b=海 c=浦 d=开 e=发 f=浦东 g=东开 h=海浦 i=开发 j=浦开 p=44444 r=浦_e s=东_e t=开_e
b a=开 b=浦 c=东 d=发 e=与 f=东开 g=开发 h=浦东 i=发与 j=东发 p=44444 r=东_b s=开_b t=发_b
e a=发 b=东 c=开 d=与 e=法 f=开发 g=发与 h=东开 i=与法 j=开与 p=44444 r=开_e s=发_e t=与_e
s a=与 b=开 c=发 d=法 e=制 f=发与 g=与法 h=开发 i=法制 j=发法 p=44444 r=发_s s=与_s t=法_s
b a=法 b=发 c=与 d=制 e=建 f=与法 g=法制 h=发与 i=制建 j=与制 p=44444 r=与_b s=法_b t=制_b
e a=制 b=与 c=法 d=建 e=设 f=法制 g=制建 h=与法 i=建设 j=法建 p=44444 r=法_e s=制_e t=建_e
b a=建 b=法 c=制 d=设 e=同 f=制建 g=建设 h=法制 i=设同 j=制设 p=44444 r=制_b s=建_b t=设_b
……
```

**Figure 3**
First Ten lines of features generated from CTB training corpus

Secondly, maximum entropy model is created from the generated features. Training was done with Gaussian prior variance of 2.5 and 1000 iterations for LBFGS parameter estimating algorithm. The LBFGS algorithm is from the C++ Maximum Entropy package (v20041229) from Le Zhang of Edinburgh University[1].

[1] http://homepages.inf.ed.ac.uk/s0450736/maxent.toolkit.html

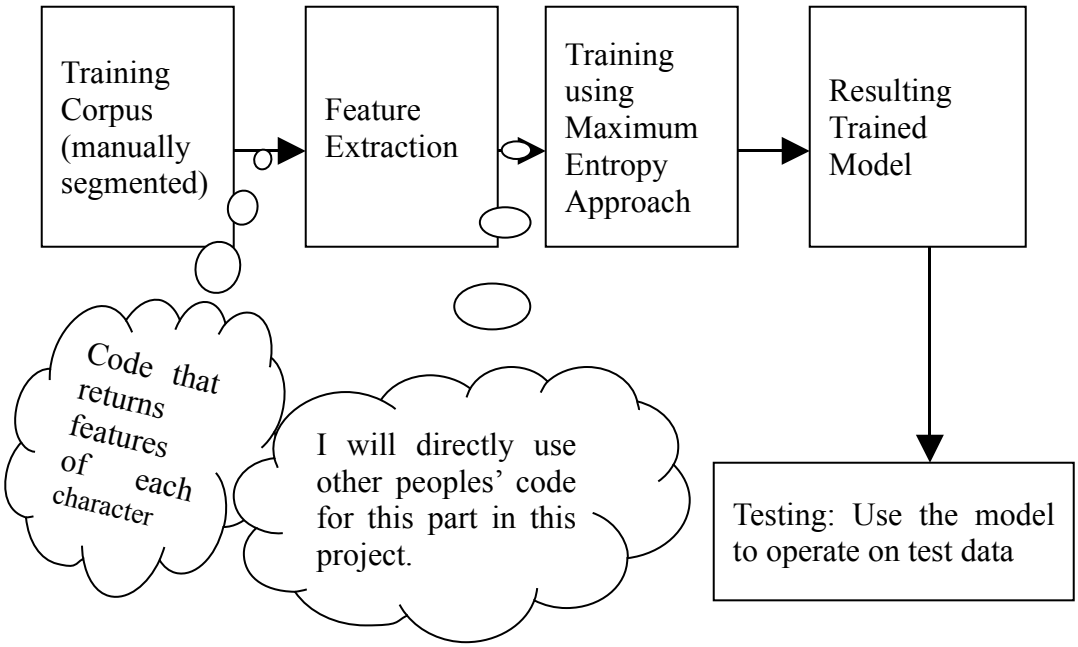
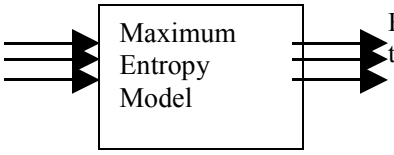



**Figure 4**
Procedure of training the original Chinese word segmenter

The resulting trained model is the original segmenter we want to obtain. It could be viewed as a black box:

1. Extracted features
2. Labeled tags




Four probabilities will be assigned to the four tags for each character

```
b a=经 b=_ c=_ d=济 e=, f=_经 g=经济 h=__ i=济, j=_济 p=44444 r=__b s=经_b t=济_b
e a=济 b=_ c=经 d=, e=决 f=经济 g=济, h=_经 i=,决 j=经, p=44444 r=经_e s=济_e t=,_e
s a=, b=经 c=济 d=决 e=战 f=济, g=,决 h=经济 i=决战 j=济决 o=1 p=44444 r=济_s s=,_s t=决_s
b a=决 b=济 c=, d=战 e=未 f=,决 g=决战 h=济, i=战未 j=,战 p=44444 r=,_b s=决_b t=战_b
e a=战 b=, c=决 d=未 e=来 f=决战 g=战未 h=,决 i=未来 j=决未 p=44444 r=决_e s=战_e t=未_e
b a=未 b=决 c=战 d=来 e=_ f=战未 g=未来 h=决战 i=来_ j=战来 p=44444 r=战_b s=未_b t=来_b
e a=来 b=战 c=未 d=_ e=_ f=未来 g=来_ h=战未 i=__ j=未_ p=44444 r=未_e s=来_e t=__e
s a=文 b=_ c=_ d=. e=李 f=_文 g=文. h=__ i=.李 j=_. p=44444 r=__s s=文_s t=._s
s a=. b=_ c=文 d=李 e=光 f=文. g=.李 h=_文 i=李光 j=文李 o=1 p=44444 r=文_s s=._s t=李_s
b a=李 b=文 c=. d=光 e=真 f=.李 g=李光 h=文. i=光真 j=.光 p=44444 r=._s s=李_s t=光_s
……
```

**Figure 5**
Features generated when ME model trained on CTB corpus is used to segment CTB test set

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| b | 0.9999854909 | e | 1.545448661e-07 | s | 1.292277283e-05 | m | 1.4 |
| 31825007e-06 | | | | | | | |
| b | 2.76948706e-07 | e | 0.9999869626 | s | 1.115367326e-05 | m | 1.6 |
| 06779292e-06 | | | | | | | |
| b | 1.75878082e-05 | e | 1.887825955e-06 | s | 0.9999798213 | m | 7.0 |
| 30401474e-07 | | | | | | | |
| b | 0.959834083 | e | 0.002633527385 | s | 0.03744575992 | m | 8.6 |
| 62964978e-05 | | | | | | | |
| b | 0.002129168957e | | 0.9685790888 | s | 0.02768034877 | m | 0.0 |
| 01611393462 | | | | | | | |
| b | 0.8396267385 | e | 0.006761076092 | s | 0.137095008 | m | 0.0 |
| 1651717746 | | | | | | | |
| b | 0.0003271738895 | | e | 0.9916017603 | s | 0.008048167245 | m |
| 2.28986012e-05 | | | | | | | |
| b | 0.05187736245 | e | 0.0003135817346 | s | 0.9475755951 | m | |
| 0.0002334607334 | | | | | | | |
| b | 0.0004363681868 | | e | 0.01006318476 | s | 0.9719487959 | m |
| 0.01755165117 | | | | | | | |
| b | 0.9881945957 | e | 0.008480184953 | s | 0.0001833554587 | m | 0.0 |

**Figure 6**
Probabilities generated when ME model trained on CTB corpus is used to segment CTB test set

## 3.4 Use of Additional Training Corpora

### 3.4.1 Basic Idea

The basic idea of optimally using additional training corpora used by Ng and Low (2006) is as follows: use the original word segmenter to segment corpora in other standards and then perform some selections on the output. Those selected segmented corpora are then added to the initial training corpus, and the enlarged training corpus will then be used for retraining.

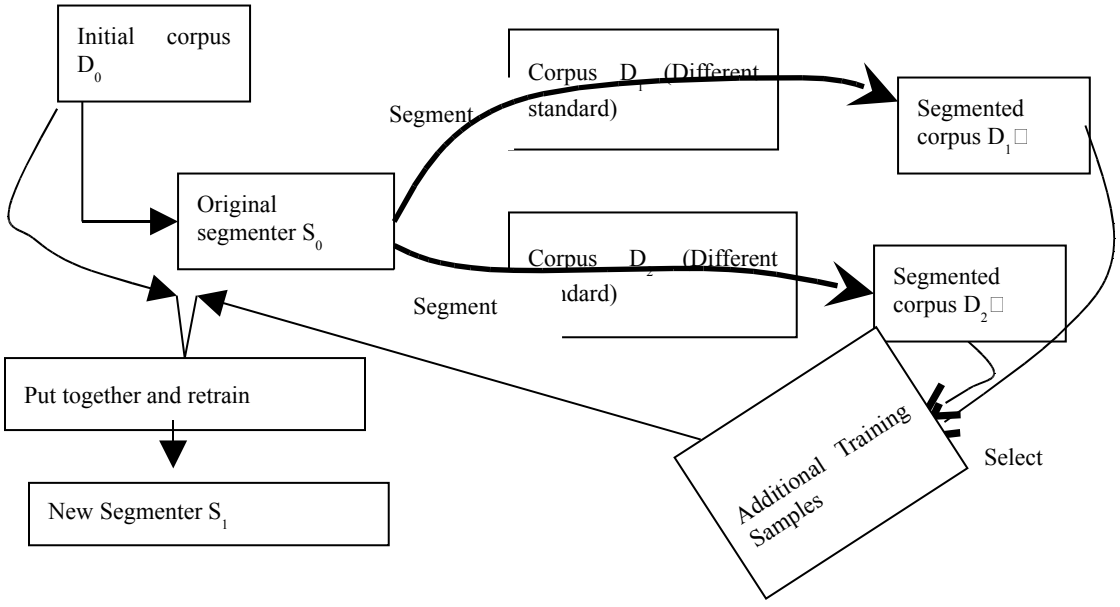



**Figure 7**
Procedure of incorporating additional training corpora

### 3.4.2 Noise Elimination and Active Learning

In order to deal with the problem of OOV words, it would be ideal if a very large training corpus can be used to train a model. But this is impossible since one single researcher or organization has limited resources and time to do the manual segmentation on the vast amount of Chinese texts, especially considering the case that new words are emerging all the time.

Therefore, it is important to make use of training corpora from other people or organization on the basis of one's original segmenter. However, adding extra training corpora is not the end of story. Additional training corpora from other organizations are segmented in different standards and naively adding them to the original corpus may introduce unwanted noise to the original data and thus corrupt it. This would decrease the segmentation accuracy, which is against our will. The noise in additional training corpora needs to be eliminated before incorporating it into the original corpus.

Besides the challenge of noise elimination, it is undesirable to add all the noise-free training data into the original corpus. This is because the training corpus in concern may grow incredibly large after incorporating many additional training corpora, and it would require longer and longer time to train the final model. This lowers the potential usability of the segmenter in the future. Therefore, it is much better if the amount of useful data added could be minimized. This is the place when active learning can be adopted to select the most useful subset of training data during the process of incorporating additional corpora.

### 3.4.3 Selecting Additional Training Examples

In order to eliminate noise in additional training corpora, we select the features of current Chinese character only when the Chinese character's assigned tag in segmented corpora, such as $D_1$' or $D_2$', is identical to its corresponding tag in gold-standard annotated corpus, such as $D_1$ or $D_2$ respectively as shown in the above diagram.

In addition of this, the assigned probability of the correct tag must be below a certain threshold to satisfy the requirement of training examples that are to be selected. This is because characters assigned with tag of too high probability are of great similarity to the existing characters in original corpus. This makes these examples a bit redundant if they are added to the original corpus. In this experiment, a threshold of 0.8 is chosen.

### 3.4.4 Algorithm

```
/*
 * Active Learning when incorporating additional training corpus
 * Select appropriate subset from extra training set of different segmentation
standards
 * Two criteria in selection: Chinese characters that are tagged correctly and
 *    have corresponding probability less than threshold theta
 * @author XiaoNi
 */
import java.io.*;
import java.util.*;

class Selector {
    static final double PROB_THRESHOLD = 0.8;
    static String growingFeat = "";
    static String otherFeat = "";
    static String segmentedOtherFeat = "";
        //features generated when using original segmenter to segment
        //training corpus of another segmentation standard
    static String segmentedOtherProb = "";
        //probabilities generated when using original segmenter to segment
        //training corpus of another segmentation standard

    public static void main (String args[]) throws Exception {

        growingFeat = args[0];
        otherFeat = args[1];
        segmentedOtherFeat = args[2];
        segmentedOtherProb = args[3];

        Vector<String> selectedFeat = new Vector();

        //set up another feature (from other standard manually segmented) reader
        FileInputStream fis1 =new FileInputStream(otherFeat);
            BufferedReader otherFeatReader =new BufferedReader(new
                                InputStreamReader(fis1));

     //set up feature (other standard segmented by original model) reader
     FileInputStream fis2 =new FileInputStream(segmentedOtherFeat);
```

```
BufferedReader segOtherFeatReader = new BufferedReader(new
InputStreamReader(fis2));

    //set up probability (other standard segmented by original model) reader
    FileInputStream fis3 =new FileInputStream(segmentedOtherProb);
      BufferedReader segOtherProbReader =new BufferedReader(new
InputStreamReader(fis3));

     //set up original feature (that is to be enlarged) writer
              BufferedWriter growFeatWriter = new BufferedWriter(new
FileWriter(growingFeat, true));

       String input;
       String segInput;
       String correctTag;
       String myModelTag;
       String myTag;   //segmented tag from prob file
       ChineseChar temp;
       while ((input = otherFeatReader.readLine())!= null)
       {
          correctTag = input.substring(0,1);
          segInput = segOtherFeatReader.readLine();
          myModelTag = segInput.substring(0, 1);
          String probInput = segOtherProbReader.readLine();
          temp = new ChineseChar(probInput);
          myTag = temp.getHighestProbTag();
          if (correctTag.equals(myTag)) {
             if (temp.getProb(correctTag) < PROB_THRESHOLD) {
                segInput = segInput.substring(1);
                segInput = correctTag.trim() + segInput;
                selectedFeat.add(segInput);
             }
          }
       }
       StringBuffer selected = new StringBuffer(selectedFeat.elementAt(0));
       for (int i = 1; i < selectedFeat.size(); i++) {
          selected.append("\n" + selectedFeat.elementAt(i));
       }
       growFeatWriter.write(selected+"\n");
        growFeatWriter.close();
    }
}
```

```
class ChineseChar
{
   double prob[];
   public ChineseChar(String predict)  //a line
   {
          prob=new double[4];  //four different tags
          StringTokenizer   st=new   StringTokenizer(predict,"   \t");

          int count=0;
          while (st.hasMoreTokens())
          {
                  st.nextToken();  //get rid of tag
                  prob[count]=Double.parseDouble(st.nextToken());

                  count++;
          }
   }
   public double getProb(String tag) throws Exception {
      char tagCopy = tag.charAt(0);
      switch (tagCopy) {
         case 'b': return prob[0];
         case 'e': return prob[1];
         case 's': return prob[2];
         case 'm': return prob[3];
         default: throw new Exception ("Invalid Tag");
      }
      }
  public String getHighestProbTag() throws Exception {
     double highest = 0;
     int highestIdx = 0;
     for (int i=0; i<4; i++) {
        if (prob[i] > highest) {
           highest = prob[i];
           highestIdx = i;
         }
      }
      switch (highestIdx) {
        case 0: return "b";
        case 1: return "e";
        case 2: return "s";
        case 3: return "m";
          default: throw new Exception ("Invalid");
      }
   }
}
```

### 3.4.5 Implementation

During the implementation phase, each bakeoff is treated as separate set of data. Within each bakeoff, the training corpus of one of the four standards is trained to obtain a maximum entropy model. This is considered the original segmenter. Later, the other three training corpora of other segmentation standards are segmented by the original segmenter. This would produce three newly-segmented text file, as well as three corresponding *.Feat files and *.Prob files, containing features generated and probabilities assigned respectively.

The "Selector.java" algorithm is then performed to incorporate the three newly-generated *.Feat files into the original feature file. The original feature file grows in size. Afterwards this enlarged feature file is trained with maximum entropy approach again to obtain a final segmenter.

The performances of original segmenter and final segmenter are then compared.

# 4 Evaluation

## 4.1 Evaluation Technique

F-measure ($\frac{2RP}{R+P}$) would be used to compute the segmentation accuracy. Inside the formula, R is the word segmentation recall which measures the total number of correct words in the output segmented by segmenter over the total number of words which were manually segmented, while P is the precision which measures the total number of correct words over the total number of words output by the segmenter.

F-measure in this experiment is computed using the official scorer used in the SIGHAN bakeoff (Emerson, 2005; Sproat and Emerson, 2003).

## 4.2 Results

The following tables list down the segmentation results between the original segmenter and the final segmenter. The header indicates on which segmentation standard the original segmenter is trained, as well as three other segmentation standards that are used as additional training corpora.

| **Bakeoff 1 Original: CTB Additional: AS, PKU, CITYU** | | |
|---|---|---|
| | Original | Final |
| Total Insertions | 1304 | 1098 |
| Total Deletions | 1150 | 1027 |
| Total Substitutions | 2783 | 2249 |

| | | |
|---|---|---|
| Total Number Change | 5237 | 4374 |
| Total True Word Count | 39922 | 39922 |
| Total Test Word Count | 40076 | 39993 |
| Total True Words Recall | 0.901 | 0.918 |
| Total Test Words Precision | 0.898 | 0.916 |
| F-measure | 0.900 | 0.917 |
| OOV Rate | 0.213 | 0.213 |
| OOV Recall Rate | 0.840 | 0.865 |
| IV Recall Rate | 0.918 | 0.932 |

**Figure 7**
Summary of segmentation results of original and final CTB segmenter

| **Bakeoff 1 Original: AS Additional: CTB, PKU, CITYU** | | |
|---|---|---|
| | Original | Final |
| Total Insertions | 105 | 105 |
| Total Deletions | 121 | 121 |
| Total Substitutions | 250 | 253 |
| Total Number Change | 476 | 479 |
| Total True Word Count | 11979 | 11979 |
| Total Test Word Count | 11963 | 11963 |
| Total True Words Recall | 0.969 | 0.969 |
| Total Test Words Precision | 0.970 | 0.970 |
| F-measure | 0.970 | 0.969 |
| OOV Rate | 1.000 | 1.000 |
| OOV Recall Rate | 0.969 | 0.969 |
| IV Recall Rate | -- | -- |

**Figure 8**
Summary of segmentation results of original and final AS segmenter

| **Bakeoff 2 Original: MSR Additional: AS, PKU, CITYU** | | |
|---|---|---|
| | Original | Final |
| Total Insertions | 1049 | 1009 |
| Total Deletions | 1117 | 1194 |
| Total Substitutions | 1945 | 1995 |
| Total Number Change | 4111 | 4198 |
| Total True Word Count | 106873 | 106873 |
| Total Test Word Count | 106805 | 106688 |
| Total True Words Recall | 0.971 | 0.970 |
| Total Test Words Precision | 0.972 | 0.972 |
| F-measure | 0.972 | 0.971 |
| OOV Rate | 0.152 | 0.152 |
| OOV Recall Rate | 0.948 | 0.948 |
| IV Recall Rate | 0.975 | 0.974 |

**Figure 9**
Summary of segmentation results of original and final MSR segmenter

| Bakeoff 2 Original: PKU Additional: AS, MSR, CITYU | | |
|---|---|---|
| | Original | Final |
| Total Insertions | 5581 | 2797 |
| Total Deletions | 2401 | 2383 |
| Total Substitutions | 8753 | 5668 |
| Total Number Change | 16735 | 10848 |
| Total True Word Count | 104372 | 104372 |
| Total Test Word Count | 107552 | 104786 |
| Total True Words Recall | 0.893 | 0.923 |
| Total Test Words Precision | 0.867 | 0.919 |
| F-measure | 0.880 | 0.921 |
| OOV Rate | 1.000 | 1.000 |
| OOV Recall Rate | 0.893 | 0.923 |
| IV Recall Rate | -- | -- |

**Figure 10**
Summary of segmentation results of original and final PKU segmenter

### 4.3 Observations

It is found that incorporating additional training corpora does not always increase the segmentation accuracy even when noise elimination and active learning are adopted during selection of additional training examples. For example, in Figure 9, the MSR model performs equally well before and after additional training corpora are added. This is similar for the AS model in Figure 8. However, in Figure 7 and Figure 10, it can be seen that the F-measure of segmenter segmenting on test data has obvious increase after incorporating additional training corpora.

With reference to Figure 1 and Figure 2, some common characteristics of CTB model and PKU model can be observed. For each of these models, the segmentation standard used for the training of original segmenter has a training corpus of relatively small size and a test data of high OOV rate. For example, CTB has a training corpus of 250K which is relatively small compared to other corpora in bakeoff 1 data, and a Test OOV of 0.181 which is relatively large compared to others.

On the other hand, AS corpus in bakeoff 1 and MSR corpus in bakeoff 2 have large training corpora and small Test OOV within their respective bakeoff data.

## 5 Exploratory Work

During the process of training a maximum entropy model, it is easy to assume that the final assigned tag to a particular Chinese character must be the same as the tag assigned with highest probability. However, discrepancy is found when the ME model trained on CTB corpus in bakeoff 1 is used to segment PKU corpus. The

following shows line 24 in the segmented feature file and segmented probability file, as well as the related section in segmented text file, after the abovementioned operation is done.

迈向 充满 希望 的 新 世纪 —— 一九九八年 新年 讲话 （ 附图片 1 张 ）

b a=图 b=( c=附 d=片 e=0 f=附图 g=图片 h=(附 i=片0 j=附片 p=44441 r=附_e s=图_e t=片_e

b 0.1849757913 e 0.3819494296 s 0.01985689736 m 0.4132178817

**Figure 11**
Text, feature and probability of PKU corpus segmented by CTB model

Such a result is arisen mainly because a dynamic programming algorithm within the maximum entropy framework is implemented. Such an algorithm is similar to the famous Viterbi algorithm (Forney, G.D., Jr, 1973) which finds the most likely sequence of hidden states. It looks for the most probable valid sequence of tags. Without this implementation, each single Chinese character would be simply assigned boundary tag with highest probability and then invalid sequences of tags such as “m” followed by “s” may occur. To prevent this, the probability of a boundary tag assignment $t_1 \ldots t_n$, given a character sequence c1…cn, is defined as follows: (Ng and Low, 2006):

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

In which $h(c_i)$ is the context of character $c_i$ defined by the maximum entropy classifier.

Therefore, during the selection of additional training examples, it is important to note that only Chinese characters which have their correct boundary tags in gold-standard annotated corpus identical to tags with highest assigned probabilities are selected. If comparison is carried out between the assigned tags in the features generated and the correct tags, the segmentation accuracy of ME model may decrease after incorporating additional training corpora.

One example used to illustrate here is incorporating PKU training corpus in bakeoff 1 into CTB training corpus. The segmentation accuracies can be seen below:

| | Original CTB segmenter | Final CTB segmenter |
|---|---|---|
| F-measure | 0.837 | 0.809 |

**Figure 11**
F-measures of original and final CTB segmenters if final assigned tags are considered in incorporating additional training corpora

One possible explanation for such a decrease in segmentation accuracy is that the additional text (e.g. PKU training corpus) contains information that "distracts" the action of the original model (e.g. CTB segmenter). This information is the actual position of Chinese character in a word in the additional text. The reason why it is considered "distracting" is that it belongs to a particular text environment and is unrelated to what the original model needs for growing. The CTB model looks through all examples in PKU training corpus and assigns boundary tag probabilities accordingly. The features generated already indicate the resulting segmented text. So it is meaningless to compare the final tag in feature file, which is already reflected in the features generated, and the gold-standard tag of a Chinese character.

In one word, we need to look for information which is common between result the original segmenter produces and correct result the organization manually produces, instead of anything else.

## 6 Conclusion

This project successfully proves that additional training corpora could be exploited to increase accuracy in Chinese word segmentation, under many restrictive conditions. These conditions include: additional training examples must be carefully selected to eliminate noise in order not to ruin the original data; active learning is adopted during the selection process to prevent redundant data from being added, for the purpose of decreasing model training time; the original corpus must have a train set with small size and a test set with high OOV rate, compared to the additional corpora used. Besides all these, during the selection, it is important to compare whether the boundary tag with highest assigned probability is identical to the correct tag of that Chinese character in gold-standard annotated corpus.

## 7 Acknowledgement

I would like to thank Dr. Ng Hwee Tou, my mentor, for introducing me to Chinese Word segmentation in NLP and motivating me with passion and insights throughout the project, Zhong Zhi, my Teaching Assistant, for guiding me with helpful suggestions and advices, as well as Prof. Lee Wee Sun, my CS2306S supervisor, for giving me the opportunity of experiencing the taste of doing research.

## 8 References


Jin Kiat Low, Hwee Tou Ng. Chinese Word Segmentation: A Maximum Entropy Approach. Unpublished Report. 2006

Keh-Jiann Chen and Shing-Huan Liu. Word identification for Mandarin Chinese sentences. In Proceedings of 14th International Conference on Computational Linguistics (COL-ING 1992), pages 101-107, 1992.

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.

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.

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

Fuchuan 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

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.

Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. Chinese word segmentation by classification of characters. In Proceedings of the Third SIGHANWorkshop on Chinese Language Processing, 2004.

Jaynes, E.T., The Maximum Entropy Formalism Where Do We Stand on Maximum Entropy?, (R.D. Levine and Myron Tribus, Eds.), The MIT Press, Cambridge, Massachusetts, pp. 15-118, 1979.

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

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.

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

Forney, G.D., Jr. The Viterbi Algorithm. Proceedings of the IEEE, pages 268-278, 1973