Computational Linguistics and Chinese Language Processing
Vol. , No. , , pp. 1

Chinese word segmentation as character tagging

Nianwen Xue*

Abstract

In this paper we report results of a supervised machine-learning approach to Chinese word segmentation. A maximum entropy tagger is trained on manually annotated data to automatically assign to Chinese characters, or *hanzi*, tags that indicate the position of a *hanzi* within a word. The tagged output is then converted into segmented text for evaluation. Preliminary results show that this approach is competitive against other supervised machine-learning segmenters reported in previous studies, achieving precision and recall rates of 95.01% and 94.94% respectively, trained on a 237K-word training set.

Key Words: Chinese word segmentation, supervised machine-learning, maximum entropy, character tagging

1. Introduction

It is generally agreed among researchers that word segmentation is a necessary first step in Chinese language processing. However, unlike English text in which sentences are sequences of words delimited by white spaces, in Chinese text, sentences are represented as strings of Chinese characters or *hanzi* without similar natural delimiters. Therefore, the first step in a Chinese language processing task is to identify the sequence of words in a sentence and mark boundaries in appropriate places. This may sound simple enough but in reality identifying words in Chinese is a non-trivial problem that has drawn a large body of research in the Chinese language processing community [Fan and Tsai, 1988; Gan, 1995; Gan, Palmer, and Lua, 1996; Guo, 1997; Jin and Chen, 1998; Sproat and Shih, 1990; Sproat *et al.*, 1996; Wu and Jiang, 1998; Wu, 2003].

It is easy to demonstrate that the lack of natural delimiters itself is not the heart of the problem. In a hypothetical language where all words are represented with a finite set of symbols, if one subset of the symbols always start a word and another subset, mutually exclusive from the previous subset, always end a word, identifying words would be a trivial

* Institute for Research in Cognitive Science , Suite 400A, 3401 Walnut Street
University of Pennsylvania, Philadelphia, PA 19104, USA
E-mail: xueniwen@linc.cis.upenn.edu

exercise. Nor can the problem be attributed to the lack of inflectional morphology. Although it is true in Indo-European languages inflectional affixes can generally be used to signal word boundaries, it is conceivable that a hypothetical language can use symbols other than inflectional morphemes to serve the same purpose. Therefore the issue is neither the lack of natural word delimiters nor the lack of inflectional morphemes in a language, rather it is whether the language has a way of unambiguously signaling the boundaries of a word.

The real difficulty in automatic Chinese word segmentation is the lack of such unambiguous word boundary indicators. In fact, most ***hanzi*** can occur in different positions within different words. The examples in Table 1 show how the Chinese character 产 ("produce") can occur in four different positions. This state of affairs makes it impossible to simply list mutually exclusive subsets of ***hanzi*** that have distinct distributions, even though the number of ***hanzi*** in the Chinese writing system is in fact finite. As long as a ***hanzi*** can occur in different word-internal positions, it cannot be relied upon to determine word boundaries as they could be if their positions were more or less fixed.

Table 1. A* hanzi *can occur in multiple word-internal positions

Position	Example
Left	产生 'to come up with'
Word by itself	产小麦 'to grow wheat'
Middle	生产线 'assembly line'
Right	生产 'to produce'

The fact that a ***hanzi*** can occur in multiple word-internal positions leads to ambiguities of various kinds, which are described in detail in [Gan, 1995]. For example, 文 can occur in both word-initial and word-final positions. It occurs in the word-final position in 日文 ("Japanese") but in the word-initial position in 文章("article"). In a sentence that has a string "日文章", as in (1)[1], an automatic segmenter would face the dilemma whether to insert a word boundary marker between 日 and 文, thus grouping 文章 as a word, or to mark 日文 as a word, to the exclusion of 章. The same scenario also applies to 章, since like 文, it can also occur in both word-initial and word-final positions.

1. (a) Segmentation I

   日文 章魚 怎麼 說?

   Japanese octopus how say

   "How to say octopus in Japanese?"

   (b) Segmentation II

[1]Adapted from [Sproat *et al.*,1996]

日 文章 鱼 怎麼 說?

Japan article fish how say

Ambiguity also arises because some ***hanzi*** should be considered to be just word components in certain contexts and words by themselves in others. For example, 鱼 can be considered to be just a word component in 章鱼. It can also be a word by itself in other contexts. Presented with the string 章鱼 in a Chinese sentence, a human or automatic segmenter would have to decide whether 鱼 should be a word by itself or form another word with the previous ***hanzi***. Given that 日, 文章, 章鱼, 鱼 are all possible words in Chinese, how does one decide that 日文 章鱼 is the right segmentation for the sentence in (1) while 日 文章 鱼 is not? Obviously it is not enough to know just what words are in the lexicon. In this specific case, a human segmenter can resort to world knowledge to resolve this ambiguity, knowing that 日 文章 鱼 would not make any kind of real-world sense.

In other cases a human segmenter can also rely on syntactic knowledge to properly segment a sentence. For instance, 枪 should be considered a word in (2a) and two words in (2b):

2. a 警察 枪-杀 了 那 个 逃犯

police gun-kill LE that CL escapee

"Police killed the escapee with a gun."

b 警察 用 枪 杀 了 那 个 逃犯

Police with gun kill LE that CL escapee

"Police killed the escapee with a gun"

In (2b), 枪 is a word by itself and forms a phrasal constituent with the preceding 用. In order to get the segmentation right for the example in (2) one needs to know, for example, that 用 has to take a complement and in the case of (2b) the complement is 枪. Therefore it is impossible for 枪 to be part of the word 枪杀. The human segmenter has little difficulty resolving these ambiguities and coming up with the correct segmentation since they have linguistic and world knowledge at their disposal. However, the means available to the human segmenter cannot be made available to computers just as easily. As a result, an automatic word segmenter would have to bypass such limitations to resolve these ambiguities.

In addition to the ambiguity problem, another problem that is often cited in the literature is the problem of so-called out-of-vocabulary or "unknown" words [Wu and Jiang, 1998]. The unknown word problem arises because machine-readable dictionaries cannot possibly list all

the words encountered in NLP tasks exhaustively[2]. For one thing, although the number of *hanzi* generally remains constant, Chinese has several productive new word creation mechanisms. First of all, new words can be created through compounding, in which new words are formed through the combination of existing words, or through *suoxie*, in which components of existing words are extracted and combined to form new words. Second, new names are created by combining existing characters in a very unpredictable manner. Third, there are also transliterations of foreign names. These are just a few of the many ways new words can be introduced in Chinese.

The key to accurate automatic word identification in Chinese lies in the successful resolution of these ambiguities and a proper way to handle out-of-vocabulary words. We have demonstrated that the ambiguities in Chinese word segmentation is due to the fact that a *hanzi* can occur in different word-internal positions. Given the proper context, generally provided by the sentence in which it occurs, the position of a *hanzi* can be determined. If the positions of all the *hanzi* in a sentence can be determined with the help of the context, the word segmentation problem would be solved. This is the line of thinking we are going to pursue in the present work. There are several reasons why we may expect this approach to work. First, Chinese words generally have fewer than four characters. As a result, the number of positions is small. Second, although each *hanzi* can in principle occur in all possible positions, not all *hanzi* behave this way. A substantial number of *hanzi* are distributed in a constrained manner. For example, 们, the plural marker, almost always occurs in the word-final position. Finally, although Chinese words cannot be exhaustively listed and new words are bound to occur in naturally occurring text, the same is not true for *hanzi*. The number of *hanzi* stays fairly constant and we do not generally expect to see new *hanzi*. In this paper, we model the Chinese word segmentation problem as a *hanzi* tagging problem and use a machine-learning algorithm to determine the word-internal positions of *hanzi* with the help of contextual information.

The remainder of this paper is organized as follows. In Section 2, we briefly review the representative approaches in the previous studies on Chinese word segmentation. In Section 3, we describe how the word segmentation problem can be modeled as a tagging problem and how the maximum entropy model is used to solve this problem. We describe our experiments in Section 4. In Section 5, we report our experimental results, using the maximum matching algorithm as a baseline. We also evaluate these results against previous approaches and discuss the contributions of different feature sets and the effectiveness of different tag sets. We conclude this paper and discuss future work in Section 6.

[2]See [Guo, 1997] for a different point of view

2. Previous Work

Various methods have been proposed to address the word segmentation problem in previous studies. Noting that linguistic information, syntactic information in particular, can help identify words, [Gan, 1995] and [Wu and Jiang, 1998] treated word segmentation as inseparable from Chinese sentence understanding as a whole. As a result, the success of the word segmentation task is tied to the success of the sentence understanding task, which is just as difficult as the word segmentation problem, if not more difficult. Most of the word segmentation systems reported in previous studies are stand-alone systems and they fall into three main categories, depending on whether they use statistical information and electronic dictionaries. These are purely statistical approaches [Sproat and Shih, 1990; Sun, Shen, and Tsou, 1998; Ge, Pratt, and Smyth, 1999; Peng and Schuurmans, 2001], non-statistical dictionary-based approaches [Liang, 1993; Gu and Mao, 1994] and statistical and dictionary-based approaches [Sproat ***et al***., 1996]. More recently work on Chinese word segmentation also includes supervised machine-learning approaches [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001].

Purely dictionary-based approaches generally addresses the ambiguity problem with some heuristics, and the most successful heuristics are variations of the maximum matching algorithm. A maximum matching algorithm is a greedy search routine that walks through a sentence trying to find the longest string of ***hanzi*** starting from a given point in the sentence that matches a word entry in a pre-compiled dictionary. For instance, assuming 关 ("close"), 心 ("heart") and 关心 ("care about") are all listed in the dictionary, given a string of ***hanzi*** 关-心, the maximum matching algorithm always favors 关心 as a word, over 关-心 as a string of two words. This is because 关心 is a longer string than 关 and both of them are in the dictionary. When the segmenter finds 关, it will continue to search and see if there is a possible extension. When it finds another word 关心 in the dictionary it will decide against inserting a word boundary between 关 and 心. When the algorithm can no longer extend the string of ***hanzi*** it stops searching and inserts a word boundary marker. The process is repeated from the next ***hanzi*** till it reaches the end of the sentence. The algorithm is successful because in a lot of cases, the longest string also happens to be correct segmentation. For example, for the example in (1), the algorithm will rightly decide that (1a) rather than (1b) is the correct segmentation for the sentence, assuming 日, 日文, 文章, 章鱼 and 鱼 are all listed in the dictionary. However, this algorithm will output the wrong segmentation for (2b), in which it will incorrectly group 枪杀 as a word. In addition, the maximum matching algorithm does not have a built-in mechanism to deal with out-of-vocabulary words. In general, the completeness of the dictionary to a large extent determines the degree of success for segmenters using this approach.

As a representative of purely statistical approaches, [Sproat and Shih, 1990] relies on the mutual information of two adjacent characters to decide whether they form a two-character word. Given a string of characters $c_1 \ldots c_n$, the pair of adjacent characters with the largest mutual information greater than a pre-determined threshold is grouped as a word. This process is repeated until there are no more pairs of adjacent characters with a mutual information value greater than the threshold. This algorithm is extended by [Sun, Shen, and Tsou, 1998] so that association measures other than mutual information are also taken into consideration. More recently, [Ge, Pratt, and Smyth, 1999; Peng and Schuurmans, 2001] applied expectation maximization methods to Chinese word segmentation. For example, [Peng and Schuurmans, 2001] used an EM-based algorithm to estimate probabilities for words in a dictionary and use mutual information to weed out proposed words whose components are not strongly associated. Purely statistical approaches have the advantage of not needing a dictionary or training data, and since unsegmented data are easy to obtain, they can be easily trained on any data source. The drawback is that statistical approaches generally do not perform well in terms of the accuracy of the segmentation.

Statistical dictionary-based approaches attempt to get the best of both worlds by combining the use of a dictionary and statistical information such as word frequency. [Sproat *et al.*, 1996] represents a dictionary as a weighted finite-state transducer. Each dictionary entry is represented as a sequence of arcs labeled with a ***hanzi*** and its phonemic transcription, starting from an initial state 0 and terminated by a ***weighted*** arch labeled with an empty string ε and a part-of-speech tag. The weight represents the estimated cost of the word, which is its negative log probability. The probabilities of the dictionary words as well as morphologically derived words not in the dictionary are estimated from a large unlabeled corpus. Given a string of acceptable symbols (all the ***hanzi*** plus the empty string), there exists a function that takes this string of symbols as input and produces as output a transducer that maps all the symbols to themselves. The path that has the cheapest cost is selected as the best segmentation for this string of characters. Compared with purely statistical approaches, statistical dictionary-based approaches have the guidance of a dictionary and as a result they generally outperform purely statistical approaches in terms of segmentation accuracy.

Recent work on Chinese word segmentation has also used the transformation-based error-driven algorithm [Brill, 1993] and achieved various degrees of success [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001]. The transformation-based error-driven algorithm is a supervised machine-learning routine first proposed by [Brill, 1993] and initially used in POS tagging as well as parsing. It has been applied to Chinese word segmentation by [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001]. Although the actual implementation of this algorithm may differ slightly, in general the transformation-based error-driven approaches try

to learn a set of n-gram rules from a training corpus and apply them to segment new text. The input to the learning routine is a (manually or automatically) segmented corpus and its unsegmenteded (or undersegmented) counterpart. The learning algorithm compares the segmented corpus and the undersegmented dummy corpus at each iteration and finds the rule that achieves the maximum gain if applied. The rule with the maximum gain is the one that makes the dummy corpus most like the reference corpus. The maximum gain is calculated with an evaluation function which quantifies the gain and takes the largest value. The rules are instantiations of a set of pre-defined templates. After the rule with the maximum gain is found, it is applied to the dummy corpus, which will better resemble the reference corpus as a result. This process is repeated until the maximum gain drops below a pre-defined threshold, which indicates improvement achieved through further training will no longer be significant. The output of the training process would be a ranked set of rules instantiating the predefined set of templates. The rules will then be used to segment new text. Like statistical approaches, this approach provides a trainable method to learn the rules from a corpus and it is not labor-intensive. The drawback is that compared with statistical approaches, this algorithm is not very efficient.

The present work represents another supervised machine-learning approach. Specifically, we applied the maximum entropy model, a statistical machine-learning algorithm to Chinese word segmentation.

3. A supervised machine-learning algorithm to Chinese word segmentation

In this section, we first formalize the idea of tagging *hanzi* based on their word-internal positions and describe the tag set we used. We then briefly describe the maximum entropy model, which has been successfully applied to POS tagging as well as parsing [Ratnaparkhi, 1996; Ratnaparkhi, 1998].

3.1 Reformulating word segmentation as a tagging problem

Before we apply the machine-learning algorithm first we convert the manually segmented words in the corpus into a tagged sequence of Chinese characters. To do this, we tag each character with one of the four tags, LL, RR, MM and LR depending on its position within a word. It is tagged LL if it occurs on the left boundary of a word, and forms a word with the character(s) on its right. It is tagged RR if it occurs on the right boundary of a word, and forms a word with the character(s) on its left. It is tagged MM if it occurs in the middle of a word. It is tagged LR if it forms a word by itself. We call such tags position-of-character (POC) tags to differentiate them from the more familiar part-of-speech (POS) tags. For example, the manually segmented string in (3a) will be tagged as (3b):

3. (a) 上海 计划 到 本 世纪 末 实现 人均 国内 生产 总值 五千 美元

(b) 上/LL 海/RR 计/LL 划/RR 到/LR 本/LR 世/LL 纪/RR 末/LR 实/LL 现/RR 人/LL 均/RR 国/LL 内/RR 生/LL 产/RR 总/LL 值/RR 五/LL 千/RR 美/LL 元/RR

(c) Shanghai plans to reach the goal of 5,000 dollars in per capita GDP by the end of the century.

Given a manually segmented corpus, a POC-tagged corpus can be derived trivially with perfect accuracy. The reason why we use such POC-tagged sequences of characters instead of applying n-gram rules to segmented corpus directly [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001] is that they are much easier to manipulate in the training process. In addition, the POC tags reflect our observation that the ambiguity problem is due to the fact that a ***hanzi*** can occur in different word-internal positions and it can be resolved in context. Naturally, while some characters have only one POC tag, most characters will receive multiple POC tags, in the same way that words can have multiple POS tags. Table 2 shows how all four of the POC tags can be assigned to the character 产 ("produce"):

Table2. A character can receive as many as four tags

Position	Tag	Example
Left	LL	产生 'to come up with'
Word by itself	LR	产 小麦 'to grow wheat'
Middle	MM	生产线 'assembly line'
Right	RR	生产 'to produce'

If there is ambiguity in segmenting a sentence or any string of ***hanzi***, then there must be some ***hanzi*** in the sentence that can receive multiple tags. For example, each of the first four characters of the sentence in (1) would have two tags. The task of the word segmentation is to choose the correct tag for each of the ***hanzi*** in the sentence. The eight possible tag sequences for (1) are shown in (4a), and the correct tag sequence is (4b).

4. (a) 日/LL|LR 文/RR|LL 章/LL|RR 鱼/RR|LR 怎/LL 么/RR 说/LR？

(b) 日/LL 文/RR 章/LL 鱼/RR 怎/LL 么/RR 说/LR？

Also like POS tags, how a character is POC-tagged in naturally occurring text is affected by the context in which it occurs. For example, if the preceding character is tagged LR or RR, then the next character can only be tagged LL or LR. How a character is tagged is also affected by the surrounding characters. For example, 关 (``close'') should be tagged RR if the previous character is 开 (``open'') and neither of them forms a word with other characters, while it should be tagged LL if the next character is 心 (``heart'') and neither of them forms a word with other characters. This state of affairs closely mimics the familiar POS tagging

problem and lends itself naturally to a solution similar to that of POS tagging. The task is one of ambiguity resolution in which the correct POC tag is determined among several possible POC tags in a specific context. Our next step is to train a maximum entropy model on the perfectly POC-tagged data derived from a manually segmented corpus to automatically POC-tag unseen text.

3.2 The maximum entropy tagger

The maximum entropy model used in POS-tagging is described in detail in [Ratnaparkhi, 1996] and the POC tagger here uses the same probability model. The probability model is defined over $H \times T$, where H is the set of possible contexts or "histories" and T is the set of possible tags. The model's joint probability of a history h and a tag t is defined as

$$p(h,t) = \pi\mu\prod_{j=1}^{k}\alpha_j^{f_j(h,t)} \quad (1)$$

where π is a normalization constant, $\{\mu, \alpha_1, ..., \alpha_k\}$ are the model parameters and $\{f_1, ..., f_k\}$ are known as features, where $f_j(h,t) \in \{0,1\}$. Each feature f_j has a corresponding parameter α_j, hat effectively serves as a "weight" of this feature. In the training process, given a sequence of characters $\{c_1, ..., c_n\}$ and their POC tags $\{t_1, ..., t_n\}$ as training data, the purpose is to determine the parameters $\{\mu, \alpha_1, ..., \alpha_k\}$ that maximize the likelihood of the training data using p:

$$L(P) = \prod_{i=1}^{n} P(h_i, t_i) = \prod_{i=1}^{n}\pi\mu\prod_{j=1}^{k}\alpha_j^{f_j(h_i,t_i)} \quad (2)$$

The success of the model in tagging depends to a large extent on the selection of suitable features. Given (h,t), a feature must encode information that helps to predict t. The features we used in this experiment are instantiations of the feature templates in (5). Feature templates (b) to (e) represent character features while (f) represents tag features. The character and tag features are also represented graphically in Figure 1, where $C_{-3}...C_3$ are characters and $T_{-3}...T_3$ are POC tags. Each arrow or arc represents one feature template. Feature template (a) represents the default feature.

5 Feature templates

(a) Default feature

(b) The current character (C_0)

(c) The previous (next) two characters (C_{-2}, C_{-1}, C_1, C_2)

(d) The previous (next) character and the current character (C_{-1} C_0, C_0 C_1),

the previous two characters (C_{-2} C_{-1}), and

the next two characters (C_1 C_2)

(e) The previous and the next character (C_{-1} C_1)

(f) The tag of the previous character (T_{-1}), and
the tag of the character two before the current character (T_{-2})

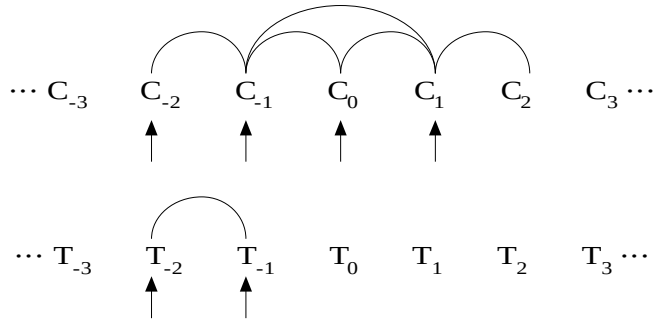

Figure 1 Features used in the maximum entropy segmenter

In general, given (h,t), these features are in the form of co-occurrence relations between t and some type of context h, or between t and some properties of the current character. For example,

$$f_i(h_i,t_i)=\begin{cases}1 & \text{if } t_{i-1}=LL \ \&\ t_i=RR \\ 0 & \text{otherwise}\end{cases}$$

This feature will map to 1 and contribute towards $p(h_i,t_i)$ if $c_{(i-1)}$ is tagged LL and c_i is tagged RR.

The feature templates in (5) encode three types of contexts. First, features based on the current and surrounding characters (5b, 5c, 5d, 5e) are extracted. Given a character in a sentence, this model will look at the current character, the previous two and next two characters. For example, if the current character is 们 (plural marker), it is very likely that it will occur as a suffix in a word, thus receiving the tag RR. On the other hand, for other characters, they might be equally likely to appear on the left, on the right or in the middle. In those cases where it occurs within a word depends on its surrounding characters. For example, if the current character is 爱 ("love"), it should perhaps be tagged LL if the next character is 护 ("protect"). However, if the previous character is 热 ("warm"), then it should perhaps be tagged RR. Second, features based on the previous tags (5f) are extracted. Information like this is useful in predicting the POC tag for the current character just as the POS tags are useful in predicting the POS tag of the current word in a similar context. For example, if the previous character is tagged LR or RR, this means that the current character must start a word, and

should be tagged either LL or LR. Finally, a default feature (5a) is used to capture cases where no other features are available. When the training is complete, the features and their corresponding parameters will be used to calculate the probability of the tag sequence of a sentence when the tagger tags unseen data. Given a sequence of characters $\{c_1,...,c_n\}$, the tagger searches for the tag sequence $\{t_1,...,t_n\}$ with the highest probability

$$P(t_1,...t_n \mid C_1,...C_n) = \prod_{i=1}^{n} P(t_i \mid h_i) \tag{3}$$

and the conditional probability of for each POC tag t given its history h is calculated as

$$P(t \mid h) = \frac{p(h,t)}{\sum_{t' \in T} P(h,t')} \tag{4}$$

4. Experiments

We conducted two experiments. In the first experiment, we used the maximum matching algorithm to establish a baseline, as comparing results across different data sources can be difficult. This experiment is also designed to test the performance of the maximum matching algorithm with or without unknown words. In the second experiment, we applied the maximum entropy model to the problem of Chinese word segmentation. The data we used is from the Penn Chinese Treebank [Xia *et al.*, 2000; Xue, Chiou, and Palmer, 2002] and it consists of Xinhua newswire articles. We took 250,389-word (426,292 characters or hanzi) worth of manually segmented data and divided them into two chunks. The first chunk has 237,791 words (404,680 Chinese characters) and is used as training data. The second chunk has 12,598 words (21,612 characters) and is held out as testing data. This data is used in both of our experiments.

4.1 Experiment One

In this experiment, we conducted two sub-experiments. In the first sub-experiment, we used a forward maximum matching algorithm to segment the testing data with a dictionary compiled from the training data. There are 497 (or 3.95%) new words (words that are not found in the training data) in the testing data. In the second sub-experiment, the same algorithm was used to segment the same testing data with a dictionary compiled from BOTH the training data and the testing data. In other words, there is no new word in the testing data.

4.2 Experiment Two

In the second experiment, a maximum entropy model was trained on a POC-tagged corpus

derived from the training data described above. In the testing phase, the sentences in the testing data were first split into sequences of *hanzi* and then tagged with this maximum entropy tagger. The tagged testing data is then converted back into word segments for evaluation. Note that converting a POC-tagged corpus into a segmented corpus is not entirely straightforward when inconsistent tagging occurs. For example, it is possible that the tagger assigns a LL-LR sequence to two adjacent characters. We made no effort to ensure the best possible conversion. The character that is POC-tagged LL is invariably combined with the following character, no matter how the latter is tagged. The example in (6) illustrates this process.

6. (a) Tagged output

在/LR 刚/LL 刚/RR 过/LL 去/RR 的/LR 一/LL 九/MM 九/MM 七/MM 年/RR ，/LR 中/LL 国/RR 进/LL 出/MM 口/RR 贸/LL 易/RR 中/LR ，/LR 国/LL 有/RR 企/LL 业/RR 与/LR 外/LL 商/RR 投/LL 资/RR 企/LL 业/RR 齐/LL 头/RR 并/LL 进/RR ，/LR 国/LL 有/RR 企/LL 业/RR 继/LL 续/RR 居/LL 于/RR 主/LL 导/RR 地/LL 位./RR ，/LR 外/LL 商/RR 投/LL 资/RR 企/LL 业/RR 仍/LL 然/RR 发/LL 挥/RR 重/LL 要/RR 的/LR 作/LL 用/RR 。/LR

(b) Segmented output

在｜刚刚｜过去｜的｜一九九七年｜，｜中国｜进出口｜贸易｜中｜，｜国有｜企业｜与｜外商｜投资｜企业｜齐头｜并进｜，｜国有｜企业｜继续｜居于｜主导｜地位｜，｜外商｜投资｜企业｜仍然｜发挥｜重要｜的｜作用｜。

(c) Gold Standard

在｜刚刚｜过去｜的｜一九九七年｜，｜中国｜进出口｜贸易｜中｜，｜国有｜企业｜与｜外商｜投资｜企业｜齐头并进｜，｜国有｜企业｜继续｜居于｜主导｜地位｜，｜外商｜投资｜企业｜仍然｜发挥｜重要｜的｜作用｜。

5. Results

In evaluating our model, we calculated both the tagging accuracy and segmentation accuracy. The calculation of the tagging accuracy is straightforward. It is simply the total number of correctly POC-tagged characters divided by the total number of characters. In evaluating segmentation accuracy, we used three measures: precision, recall and balanced F-score. Precision p is defined as the number of correctly segmented words divided by the total number of words in the automatically segmented corpus. Recall r is defined as the number of correctly segmented words divided by the total number of words in the gold standard, which is the manually annotated corpus. F-score f is defined as follows:

$$f = \frac{p \times r \times 2}{p + r} \tag{5}$$

The results of the three experiments are tabulated in Table 3:

Table 3. Experimental results

	Tagging accuracy		*Segmentation accuracy*			
Experiments	Training	Testing	Testing			
			p(%)	r(%)	f(%)	r(% new words)
1a	n/a	n/a	87.34	92.34	89.77	1.37
1b	n/a	n/a	94.51	95.80	95.15	n/a
2	97.90	96.05	95.01	94.94	94.98	70.20

The results from Experiment One show that the accuracy of the maximum matching algorithm degrades sharply when there are new words in the testing data, even when there is only a small proportion of them. Assuming an ideal scenario where there is no new word in the testing data, the maximum matching algorithm achieves an F-score of 95.15%. However, when there are new words (words not found the training data), the accuracy drops to only 89.77% in F-score. In contrast, the maximum entropy tagger achieves an accuracy of 94.98% by the balanced F-score even when there are new words in testing data. This result is only slightly lower than the 95.15% that the maximum matching algorithm achieves when there is no new word. An analysis of the new words (words not in the training data) is more revealing. Of the 510 words that are found in the testing data but not in the training data, 7 or 1.37% of them are correctly segmented by the maximum matching algorithm (Experiment 1a), while the maximum entropy model correctly segmented 70.20%, or 358 of them. The 7 words the maximum matching algorithm segmented correctly happen to be single-character words. This is expected because the maximum matching algorithm stops when it can no longer extend a string of ***hanzi*** based on a dictionary. In contrast, for the maximum entropy model, unknown words are predicted based on the distribution of their components. Even though the new words are not found in the training data, their components can still be found and words can be proposed based on the distribution of their components, a property that is typical of back-off statistical models. The fact the recall of the unknown words is well below the overall recall suggests that statistics of the unknown words are harder to collect than the known words.

The results of this segmenter against previous studies are harder to assess. One reason why this is difficult is that the accuracy representing segmenter performance can only be meaningfully interpreted if there is a widely accepted definition of wordhood in Chinese. It has been well-documented in the linguistics literature [Dai, 1992; Packard, 2000; Xue, 2001] that phonological, syntactic and semantic criteria do not converge to allow a single notion of "word" in Chinese. In practice, noting the difficulty in defining wordhood, researchers in

automatic word segmentation of Chinese text generally adopt their own working definitions of what a word is, or simply rely on native speakers' subjective judgments. The problem with native speakers' subjective judgements is that native speakers generally show great inconsistency in their judgments of wordhood, as should perhaps be expected given the difficulty of defining what a word is in Chinese. For example, Wu and Fung [1994] introduced an evaluation method which they call nk-blind. To deal with the inconsistency they proposed a scheme in which n human judges are asked to segment a text independently. They then compare the segmentation of an automatic segmenter with those of the human judges. For a given "word" produced by the automatic segmenter, there may be k human judges agreeing that this is a word, where k is between $zero$ and n. For eight human judges, the precision of the segmentation with which all the human judges agree is only 30%, while the precision of the segmentation that at least one human judge agrees with is 90%. [Sproat *et al.*, 1996] adopted a different evaluation method since their work on Chinese word segmentation is tailored for use in a text-to-speech system. Their subjects, who have no training in linguistics, are instructed to segment sentences by marking all the places they might be plausibly pause if they were reading the text aloud. They tested inter-subject consistency on six native speakers of Mandarin Chinese and the average inter-subject consistency is 76%. These experiments attest the difficulty of evaluating the performance of different segmenters.

The situation is improving with the emergence of published segmentation standards and corpora manually segmented in keeping with these standards [Xia, 2000; Yu *et al.*,1998; CKIP, 1995]. Still, the corpora can vary by size, the complexity of the sentences in the corpora, so on and so forth. Unless the segmenters are tested with a single standard corpus, the performance of different segmenters are still hard to gauge. Still some preliminary observations can be made in this regard. Our accuracy is much higher that those reported in [Hockenmaier and Brew, 1998] and [Xue, 2001], who used error-driven transformation-based learning to learn a set of n-gram rules to do a series of merge and split operations on data from Xinhua news, the same data source as that of ours. The results they reported are 87.9% (trained on 100,000 words) and 90.2% (trained on 80,000 words) respectively, measured by the balanced F-score. Using a statistical model called prediction by partial matching (PPM), Teahan *et al*. [2000] reported a significantly better result. The model was trained on a million words from Guo Jin's Mandarin Chinese PH corpus and tested on five 500-segment files. The reported F-scores are in a range between 89.4% and 98.6%, averaging 94.4%. Since the data is also from Xinhua newswire, some comparison can be made between our results and this model. With less training data, our results using the maximum entropy model are slightly higher (by 0.48%). Tested on the same test data as ours, the Microsoft system [Wu, 2003] achieved a higher accuracy, achieving precision and recall rates of 95.98% and 96.36%

respectively, using a dictionary of around 89K words, compared with around 19K unique words in our training data. We believe our approach can achieve higher accuracy with more training data.

5.1 Personal names

It has long been noted that personal names often pose a serious problem for automatic word segmentation, presumably because new names are constantly made up and it is impossible to list them exhaustively in pre-compiled dictionaries that dictionary-based approaches heavily rely on. It is expected that these names should not generally be a problem for the present character-based approach in the same way because new words are not distinct problems for this approach. Among the 137 personal names (122 unique names, both Chinese names and foreign name transliterations) found in the testing data, 119 of them are segmented correctly, with a recall of 86.86%. The 18 wrongly segmented names are given in Table 4. In general, longer names, especially foreign names, are more likely to cause problems for this model.

Table 4. Incorrectly segmented personal names

Correct Segmentation	Segmenter Output
穆罕默德．胡期尼．穆巴拉克	穆罕｜默德．胡期尼．穆巴拉克
加央多吉	加央｜多｜吉
袁养和	袁养｜和
汪家(廖去广加金旁)	汪｜家｜(｜廖去｜广加金旁)
桑普拉斯	桑普拉斯伤愈
彭定康	彭定｜康道别
黄河明	黄河｜明以
顾明	顾明、
金硕仁	金硕｜仁
克里斯蒂娜．斯米贡	克里斯蒂娜．斯米贡。
江｜主席	江主席
米本育代	米｜本育代
中屋朱美	中屋｜朱美
里戈韦塔．门楚	家里戈韦塔．门楚
凯基特．荷布南南德	凯基特．荷布｜南南德
王咸儒	王咸儒说
里库佩罗	里库｜佩罗 23 日
令狐道成	令｜狐道成

5.2 Contribution of Features

In an effort to assess the effectiveness of the different types of features, we retrained our system by taking out each group of features in (5). The most effective features are the ones which, when not used, result in the most loss in accuracy. Table 5 shows that there is loss of accuracy when any of the six groups of features are not used. This means that all of the features in (5) made a positive contribution to the overall model. It is also clear that the features in (5d), which are pairs of Chinese characters, are the most effective. A substantial number of the features in (5d) encode two-character words and are thus good indicators of how the current character should be tagged. For example, if C_0 = 功 and C_1 = 能, this is good indication that 功 will start a word 功能, thus receive a tag LL. The previous (next) two characters individually (5c), the previous tags (5f) and the current character (5b) also made a substantial contribution to the overall model. The least useful features are the previous and the next character together (5e) and the default feature. The default feature is useful when no other features are invoked, e.g. when the current character is unknown and the previous two and next two characters are also unknown. It is not as effective as other features presumably because the likelihood of this scenario happening is small, given the characters in Chinese are limited in number.

Table 5. The effectiveness of different features

Features	*Tagging accuracy*	*Segmentation accuracy*		
		p(%)	r(%)	f(%)
all	96.05	95.01	94.94	94.98
w/o (a)	96.03	94.97	94.94	94.96
w/o (e)	95.92	94.85	94.86	94.85
w/o (b)	95.16	93.99	93.95	93.97
w/o (f)	95.41	93.88	93.95	93.91
w/o (c)	95.11	93.40	93.95	93.67
w/o (d)	92.62	91.04	91.06	91.05

5.3 Effects of Tag Sets

The choice of our POC tag set is based on linguistic intuitions. The use of four tags is linguistically intuitive in that LL tags morphemes that are prefixes or stems in the absence of prefixes, RR tags morphemes that are suffixes or stems in the absence of suffixes, MM tags stems with affixes and LR tags stems without affixes. The results in Table 6 show that our linguistically intuitive tag set is also the most effective. The use of three tags (LL for beginning of a word, RR for continuation of a word and LR for word by itself) that has been proven to be the most useful for baseNP chunking [Ramshaw and Marcus, 1995] results in comparable performance in segmentation accuracy. The use of two tags (LL for beginning of

a word and RR otherwise) results in substantial loss in segmentation accuracy while gaining in tagging accuracy. This is a somewhat surprising result since there is no inconsistent tagging with this tag set and thus no loss in accuracy in the post-tagging conversion process.

Table 6. The effectiveness of different tagsets

Tagset	*Tagging accuracy*	*Segmentation accuracy*		
		p(%)	r(%)	f(%)
Two	97.51	94.37	94.40	94.38
Three	96.51	95.09	94.83	94.96
Four	96.05	95.01	94.94	94.98

6. Conclusions and Future Work

The preliminary results show that the maximum entropy model can be effectively applied to Chinese word segmentation. It is more robust than the maximum matching algorithm in the sense that it can handle unknown words much more effectively. The results also show that our approach is competitive against other machine-learning models.

Much work needs to be done to evaluate this approach more thoroughly. For example, more experiments need to be performed on data sources other than the newswire type and on standards other than the Penn Chinese Treebank. In addition, we plan to explore ways to further improve this segmenter. For instance, we expect that the segmenter accuracy can still be improved as more training data become available. Refined pre-processing or post-processing steps could also help improve segmentation accuracy. For example, instead of tagging ***hanzi*** directly it might be possible to tag morphemes, which may or may not be composed of just one ***hanzi***. There might also be better ways to convert a tagged sequence into a word sequence than the simple approach we adopted.

Acknowledgement

I would like to thank Susan Converse for her detailed comments on a previous version of this work. The comments of the three anonymous reviewers also led to significant improvement of this paper. I would also like to thank Andi Wu for for his help in evaluating the results reported here and Richard Sproat, the guest editor of this special issue, for help with references. All inadequacies that still exist in this work are my own, of course. This research was funded by DARPA N66001-00-1-8915.

References

Brill, Eric, 1993. *A Corpus-based Approach to Language Learning*. Ph.D. thesis, University of Pennsylvania.

CKIP, 1995. An Introduction to the Academia Sinica Balanced Corpus (in Chinese). Technical Report 95-02, Taipei: Academia Sinica.

Dai, Xiang-Ling, 1992. *Chinese Morphology and its Interface with the Syntax*. Ph.D. thesis, Ohio State University.

Fan, C. K. and W. H. Tsai, 1988. ‘‘Automatic word identification in Chinese sentences by the relaxation technique”. *Computer Processing of Chinese and Oriental Languages*, 4(1):33–56.

Gan, Kok-Wee, 1995. *Integrating Word Boundary Disambiguation with Sentence Understanding* . Ph.D. thesis, National University of Singapore.

Gan, Kok-Wee, Martha Palmer, and Kim-Teng Lua, 1996. ‘‘A statistically emergent approach for language processing: Application to modeling context effects in ambiguous Chinese word boundary perception”. *Computational Linguistics*, 22(4):531–53.

Ge, Xianping, Wanda Pratt, and Padhraic Smyth, 1999. ‘‘Dicovering Chinese words from unsegmented text”. In *SIGIR’99*.

Gu, Ping and Yuhang Mao, 1994. ‘‘Hanyu zidong fenci de jinlin pipei suanfa jiqi zai qhfy hanying jiqi fanyi xitong zhong de shixian”. [the adjacent matching algorithm of Chinese automatic word segmentation and its implementation in the qhfy Chinese-english system. In *International Conference on Chinese Computating*. Singapore.

Guo, Jin, 1997. ‘‘Critical tokenization and its properties”. *Computational Linguistics*, 23(4):569–596.

Hockenmaier, Julia and Chris Brew, 1998. ‘‘Error-driven segmentation of Chinese”. *Communications of COLIPS*, 1(1):69–84.

Jin, Wangying and Lei Chen, 1998. ‘‘Identifying unknown words in Chinese corpora”. In *The First Workshop on Chinese Language Processing*. University of Pennsylvania, Philadelphia.

Liang, Nanyuan, 1993. ‘‘shumian hanyu zidong fenci xitong cdws”. *Journal of Chinese Information Processing*, 1(1):44–52.

Packard, Jerome, 2000. *The Morphology of Chinese: A Linguistics and Cognitive Approach*. Cambridge: Cambridge University Press.

Palmer, David, 1997. ‘‘A trainable rule-based algorithm to word segmentation”. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics*. Madrid, Spain.

Peng, Fuchun and Dale Schuurmans, 2001. “Self-supervised Chinese word segmentation”. In F. Hoffman *et al* (ed.), *Advances in Intelligent Data Analysis, Proceedings of the Fourth International Conference (IDA-01)*. Heidelberg. Springer-Verlag.

Ramshaw, Lance and Mitchell P. Marcus, 1995. ‘‘Text chunking using transformation-based learning”. In *Proceedings of the Third ACL Workshop on Very Large Corpora*.

Ratnaparkhi, Adwait, 1996. ‘‘A maximum entropy part-of-speech tagger”. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*. University of Pennsylvania.

Ratnaparkhi, Adwait, 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Sproat, R. and C. L. Shih, 1990. ‘‘A statistical method for finding word boundaries in Chinese text”. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351.

Sproat, R., Chilin Shih, William Gale, and Nancy Chang, 1996. ‘‘A stochastic finite-state word-segmentation algorithm for Chinese”. *Computational Linguistics*, 22(3):377–404.

Sun, Maosong, Dayang Shen, and Benjamin K. Tsou, 1998. ‘‘Chinese word segmentation without using lexicon and hand-crafted training data”. In *Proceedings of COLING-ACL’98.*

Wu, Andi, this issue. ‘‘Customizable segmentation of morphologically derived words in Chinese”. *Computational Linguistics and Chinese Language Processing*.

Wu, Andi and Zixin Jiang, 1998. ‘‘Word segmentation in sentence analysis”. In *Proceedings of the 1998 International Conference on Chinese Information Processing*. Beijing, China.

Wu, Dekai and Pascale Fung, 1994. ‘‘Improving Chinese tokenization with linguistic filters on statistical lexical acquisition”. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*.

Xia, Fei, 2000. The Segmentation Guidelines for Chinese Treebank Project. Technical Report IRCS 00-06, University of Pennsylvania.

Xia, Fei, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus, 2000. “Developing Guidelines and Ensuring Consistency for Chinese Text Annotation”. In *Proc. of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*. Athens, Greece.

Xue, Nianwen, 2001. *Defining and Automatically Identifying Words in Chinese*. Ph.D. thesis, University of Delaware.

Xue, Nianwen, Fu-Dong Chiou, and Martha Palmer, 2002. ‘‘Building a large annotated Chinese corpus”. In *The Proceedings of the 19th International Conference on Computational Linguistics*. Taipei, Taiwan.

Yu, Shiwen, Xuefeng Zhu, Hui Wang, and Yunyun Zhang, 1998. *The Grammatical Knowledge-base of Contemporary Chinese — A Complete Specification (in Chinese)*. Tsinghua University Press.