Unsupervised Segmentation of Chinese Text by Use of Branching Entropy

Zhihui Jin and **Kumiko Tanaka-Ishii**
Graduate School of Information Science and Technology
University of Tokyo

Abstract

We propose an unsupervised segmentation method based on an assumption about language data: that the increasing point of entropy of successive characters is the location of a word boundary. A large-scale experiment was conducted by using 200 MB of unsegmented training data and 1 MB of test data, and precision of 90% was attained with recall being around 80%. Moreover, we found that the precision was stable at around 90% independently of the learning data size.

1 Introduction

The theme of this paper is the following assumption:

> The uncertainty of tokens coming after a sequence helps determine whether a given position is at a boundary. (A)

Intuitively, as illustrated in Figure 1, the variety of successive tokens at each character inside a word monotonically decreases according to the offset length, because the longer the preceding character n-gram, the longer the preceding context and the more it restricts the appearance of possible next tokens. For example, it is easier to guess which character comes after "natura" than after "na". On the other hand, the uncertainty at the position of a word border becomes greater, and the complexity increases, as the position is out of context. With the same example, it is difficult to guess which character comes after "natural ". This suggests that a word border can be detected by focusing on the differentials of the uncertainty of branching.

In this paper, we report our study on applying this assumption to Chinese word segmentation by formalizing the uncertainty of successive tokens via the branching entropy (which we mathematically define in the next section). Our intention in this paper is above all to study the fundamental and scientific statistical property underlying language data, so that it can be applied to language engineering.

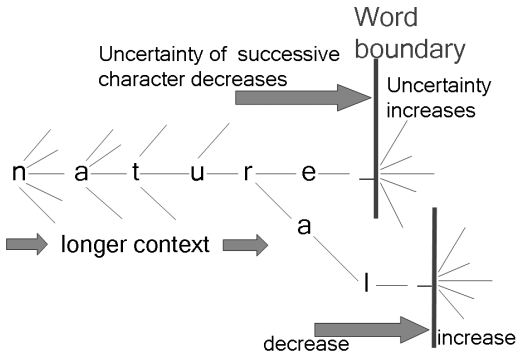

Figure 1: Intuitive illustration of a variety of successive tokens and a word boundary

The above assumption (A) dates back to the fundamental work done by Harris (Harris, 1955), where he says that when the number of different tokens coming after every prefix of a word marks the maximum value, then the location corresponds to the morpheme boundary. Recently, with the increasing availability of corpora, this property underlying language has been tested through segmentation into words and morphemes. Kempe (Kempe, 1999) reports a preliminary experiment to detect word borders in German and English texts by monitoring the entropy of successive characters for 4-grams. Also, the second author of this paper (Tanaka-Ishii, 2005) have shown how Japanese and Chinese can be segmented into words by formalizing the uncertainty with the branching entropy. Even though the test data was limited to a small amount in this work, the report suggested how assumption

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 428–435,
Sydney, July 2006.

(A) holds better when each of the sequence elements forms a semantic unit. This motivated our work to conduct a further, larger-scale test in the Chinese language, which is the only human language consisting entirely of ideograms (i.e., semantic units). In this sense, the choice of Chinese as the language in our work is essential.

If the assumption holds well, the most important and direct application is unsupervised text segmentation into words. Many works in unsupervised segmentation so far could be interpreted as formulating assumption (A) in a similar sense where branching stays low inside words but increases at a word or morpheme border. None of these works, however, is directly based on (A), and they introduce other factors within their overall methodologies. Some works are based on in-word branching frequencies formulated in an original evaluation function, as in (Ando and Lee, 2000) (boundary precision=84.5%,recall=78.0%, tested on 12500 Japanese ideogram words). Sun et al. (Sun et al., 1998) uses mutual information (boundary p=91.8%, no report for recall, 1588 Chinese characters), and Feng(Feng et al., 2004) incorporates branching counts in the evaluation function to be optimized for obtaining boundaries (word precision=76%, recall=78%, 2000 sentences). From the performance results listed here, we can see that unsupervised segmentation is more difficult, by far, than supervised segmentation; therefore, the algorithms are complex, and previous studies have tended to be limited in terms of both the test corpus size and the target.

In contrast, as assumption (A) is simple, we keep this simplicity in our formalization and directly test the assumption on a large-scale test corpus consisting of 1001 KB manually segmented data with the training corpus consisting of 200 MB of Chinese text.

Chinese is such an important language that supervised segmentation methods are already very mature. The current state-of-the-art segmentation software developed by (Low et al., 2005), which ranks as the best in the SIGHAN bakeoff (Emerson, 2005), attains word precision and recall of 96.9% and 96.8%, respectively, on the PKU track. There is also free

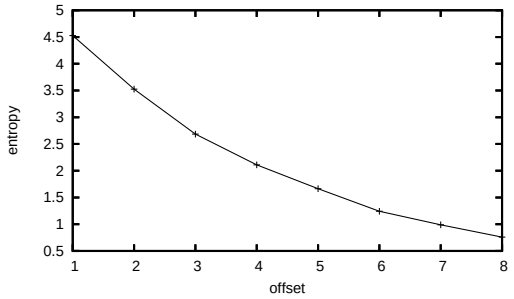

Figure 2: Decrease in $H(X|X_n)$ for Chinese characters when n is increased

software such as (Zhang et al., 2003) whose performance is also high. Even then, as most supervised methods learn on manually segmented newspaper data, when the input text is not from newspapers, the performance can be insufficient. Given that the construction of learning data is costly, we believe the performance can be raised by combining the supervised and unsupervised methods.

Consequently, this paper verifies assumption (A) in a fundamental manner for Chinese text and addresses the questions of why and to what extent (A) holds, when applying it to the Chinese word segmentation problem. We first formalize assumption (A) in a general manner.

2 The Assumption

Given a set of elements χ and a set of n-gram sequences χ_n formed of χ, the conditional entropy of an element occurring after an n-gram sequence X_n is defined as

$$H(X|X_n) = -\sum_{x_n \in \chi_n} P(x_n) \sum_{x \in \chi} P(x|x_n) \log P(x|x_n), \quad (1)$$

where $P(x) = P(X = x)$, $P(x|x_n) = P(X = x|X_n = x_n)$, and $P(X = x)$ indicates the probability of occurrence of x.

A well-known observation on language data states that $H(X|X_n)$ decreases as n increases (Bell et al., 1990). For example, Figure 2 shows how $H(X|X_n)$ shifts when n increases from 1 to 8 characters, where n is the length of a word prefix. This is calculated for all words existing in the test corpus, with the entropy being measured in the learning data (the learning and test data are defined in §4).

This phenomenon indicates that X will become easier to estimate as the context of X_n

gets longer. This can be intuitively understood: it is easy to guess that "e" will follow after "Hello! How ar", but it is difficult to guess what comes after the short string "He". The last term $-\log P(x|x_n)$ in the above formula indicates the information of a token of x coming after x_n, and thus the branching after x_n. The latter half of the formula, the local entropy value for a given x_n,

$$H(X|X_n = x_n) = -\sum_{x \in \chi} P(x|x_n) \log P(x|x_n), \quad (2)$$

indicates the average information of branching for a *specific* n-gram sequence x_n. As our interest in this paper is this local entropy, we denote $H(X|X_n = x_n)$ simply as $h(x_n)$ in the rest of this paper.

The decrease in $H(X|X_n)$ globally indicates that given an n-length sequence x_n and another $(n+1)$-length sequence y_{n+1}, the following inequality holds *on average*:

$$h(x_n) > h(y_{n+1}). \quad (3)$$

One reason why inequality (3) holds for language data is that there is *context* in language, and y_{n+1} carries a *longer context* as compared with x_n. Therefore, if we suppose that x_n is the prefix of x_{n+1}, then it is very likely that

$$h(x_n) > h(x_{n+1}) \quad (4)$$

holds, because the longer the preceding n-gram, the longer the *same* context. For example, it is easier to guess what comes after x_6="natura" than what comes after x_5 = "natur". Therefore, the decrease in $H(X|X_n)$ can be expressed as the concept that if the context is longer, the uncertainty of the branching decreases on average. Then, taking the logical contraposition, if the uncertainty does not decrease, the context is not longer, which can be interpreted as the following:

> If the entropy of successive tokens increases, the location is at a context border. (B)

For example, in the case of x_7 = "natural", the entropy h("natural") should be larger than h("natura"), because it is uncertain what character will allow x_7 to succeed. In the next section, we utilize assumption (B) to detect context boundaries.

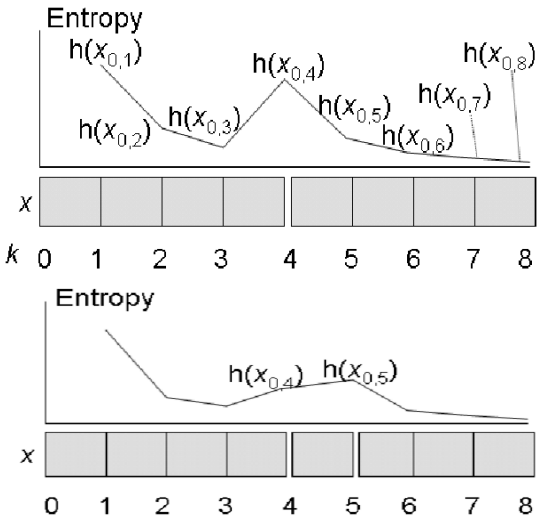

Figure 3: Our model for boundary detection based on the entropy of branching

3 Boundary Detection Using the Entropy of Branching

Assumption (B) gives a hint on how to utilize the branching entropy as an indicator of the context boundary. When two semantic units, both longer than 1, are put together, the entropy would appear as in the first figure of Figure 3. The first semantic unit is from offsets 0 to 4, and the second is from 4 to 8, with each unit formed by elements of χ. In the figure, one possible transition of the branching degree is shown, where the plot at k on the horizontal axis denotes the entropy for $h(x_{0,k})$ and $x_{n,m}$ denotes the substring between offsets n and m.

Ideally, the entropy would take a maximum at 4, because it will decrease as k is increased in the ranges of $k < 4$ and $4 < k < 8$, and at $k = 4$, it will rise. Therefore, the position at $k = 4$ is detected as the "local maximum value" when monitoring $h(x_{0,k})$ over k. The boundary condition after such observation can be redefined as the following:

B$_{max}$ Boundaries are locations where the entropy is locally maximized.

A similar method is proposed by Harris (Harris, 1955), where morpheme borders can be detected by using the local maximum of the number of different tokens coming after a prefix.

This only holds, however, for semantic units longer than 1. Units often have a length of

1, especially in our case with Chinese characters as elements, so that there are many one-character words. If a unit has length 1, then the situation will look like the second graph in Figure 3, where three semantic units, $x_{0,4}$, $x_{4,5}$, and $x_{5,8}$, are present, with the middle unit having length 1. First, at $k = 4$, the value of h increases. At $k = 5$, the value may increase or decrease, because the longer context results in an uncertainty decrease, *though an uncertainty decrease does not necessarily mean a longer context.* When h increases at $k = 5$, the situation will look like the second graph. In this case, the condition B_{max} will not suffice, and we need a second boundary condition:

$\mathrm{B}_{increase}$ Boundaries are locations where the entropy is increased.

On the other hand, when h decreases at $k = 5$, then even $\mathrm{B}_{increase}$ cannot be applied to detect $k = 5$ as a boundary. We have other chances to detect $k = 5$, however, by considering $h(x_{i,k})$, where $0 < i < k$. According to inequality (3), then, a similar trend should be present for plots of $h(x_{i,k})$, assuming that $h(x_{0,n}) > h(x_{0,n+1})$; then, we have

$$h(x_{i,n}) > h(x_{i,n+1}), \quad \text{for} \quad 0 < i < n. \qquad (5)$$

The value $h(x_{i,k})$ would hopefully rise for some i if the boundary at $k = 5$ is important, although $h(x_{i,k})$ can increase or decrease at $k = 5$, just as in the case for $h(x_{0,n})$.

Therefore, when the target language consists of many one-element units, $\mathrm{B}_{increase}$ is crucial for collecting all boundaries. Note that the boundaries detected by B_{max} are included in those detected by the condition $\mathrm{B}_{increase}$, and also that $\mathrm{B}_{increase}$ is a boundary condition representing the assumption (B) more directly.

So far, we have considered only regular-order processing: the branching degree is calculated for *successive* elements of x_n. We can also consider the reverse order, which involves calculating h for the *previous* element of x_n. In the case of the previous element, the question is whether the head of x_n forms the *beginning* of a context boundary.

Next, we move on to explain how we actually applied the above formalization to the problem of Chinese segmentation.

4 Data

The whole data for training amounted to 200 MB, from the Contemporary Chinese Corpus of the Center of Chinese Linguistics at Peking University (Center for Chinese Linguistics, 2006). It consists of several years of Peoples' Daily newspapers, contemporary Chinese literature, and some popular Chinese magazines. Note that as our method is unsupervised, this learning corpus is just text without any segmentation.

The test data were constructed by selecting sentences from the manually segmented People's Daily corpus of Peking University. In total, the test data amounts to 1001 KB, consisting 147026 Chinese words. The word boundaries indicated in the corpus were used as our golden standard.

As punctuation is clear from text boundaries in Chinese text, we pre-processed the test data by segmenting sentences at punctuation locations to form text fragments. Then, from all fragments, n-grams of less than 6 characters were obtained. The branching entropies for all these n-grams existing within the test data were obtained from the 200 MB of data.

We used 6 as the maximum n-gram length because Chinese words with a length of more than 5 characters are rare. Therefore, scanning the n-grams up to a length of 6 was sufficient. Another reason is that we actually conducted the experiment up to 8-grams, but the performance did not improve from when we used 6-grams.

Using this list of words ranging from unigrams to 6-grams and their branching entropies, the test data were processed so as to obtain the word boundaries.

5 Analysis for Small Examples

Figure 4 shows an actual graph of the entropy shift for the input phrase 未来发展的目标和指导 方针 (*wei lai fa zhan de mu biao he zhi dao fang zhen*, the aim and guideline of future development). The upper figure shows the entropy shift for the forward case, and the lower figure shows the entropy shift for the backward case. Note that for the backward case, the branching entropy was calculated for characters *before* the x_n.

In the upper figure, there are two lines, one

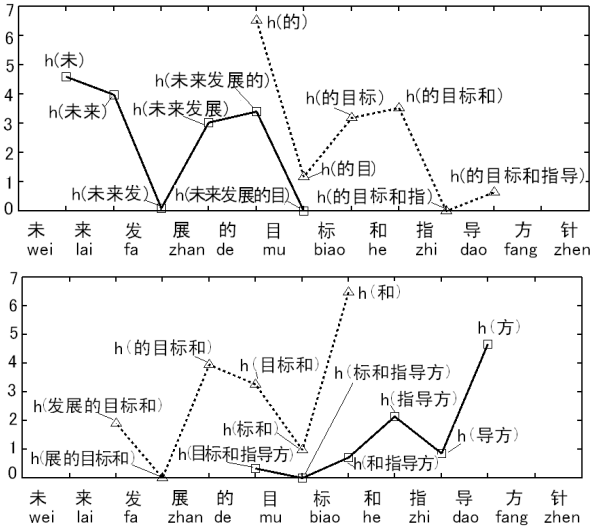

Figure 4: Entropy shift for a small example (forward and backward)

for the branching entropy after the substrings starting from 未. The leftmost line plots h(未), h(未来) ... h(未来发展的目). There are two increasing points, indicating that the phrase was segmented between 发展 and 的, and between 的 and 目标. The second line plots h(的) ...h(的目标和指导). The increasing locations are between 目标 and 和, between 和 and 指导, and after 指导.

The lower figure is the same. There are two lines, one for the branching entropy before the substring ending with suffix 方. The rightmost line plots h(方), h(导方) ...h(目标和指导方) running from back to front. We can see increasing points (as seen from back to front) between 和 and 指导, and between 的 and 目标. As for the last line, it also starts from 目 and runs from back to front, indicating boundaries between 的 and 目标, between 发展 and 的, and just before 发展.

If we consider all the increasing points in all four lines and take the set union of them, we obtain the correct segmentation as follows:

未来|发展|的| 目标|和|指导|方针,

which is the 100 % correct segmentation in terms of both recall and precision.

In fact, as there are 12 characters in this input, there should be 12 lines starting from each character for all substrings. For readability, however, we only show two lines each for the forward and backward cases. Also, the maximum length of a line is 6, because we only took 6-grams out of the learning data. If we consider all the increasing points in all 12 lines and take the set union, then we again obtain 100 % precision and recall. It is amazing how all 12 lines indicate only correct word boundaries.

Also, note how the correct full segmentation is obtained only with partial information from 4 lines taken from the 12 lines. Based on this observation, we next explain the algorithm that we used for a larger-scale experiment.

6 Algorithm for Segmentation

Having determined the entropy for all n-grams in the learning data, we could scan through each chunk of test data in both the forward order and the backward order to determine the locations of segmentation.

As our intention in this paper is above all to study the innate linguistic structure described by assumption (B), we do not want to add any artifacts other than this assumption. For such exact verification, we have to scan through all possible substrings of an input, which amounts to $O(n^2)$ computational complexity, where n indicates the input length of characters.

Usually, however, $h(x_{m,n})$ becomes impossible to measure when $n - m$ becomes large. Also, as noted in the previous section, words longer than 6 characters are very rare in Chinese text. Therefore, given a string x, all n-grams of no more than 6 grams are scanned, and the points where the boundary condition holds are output as boundaries.

As for the boundary conditions, we have B_{max} and $\mathrm{B}_{increase}$, and we also utilize $\mathrm{B}_{ordinary}$, where location n is considered as a boundary when the branching entropy $h(x_n)$ is simply above a given threshold. Precisely, there are three boundary conditions:

$\mathbf{B}_{max}$ $h(x_n) > valmax$,
where $h(x_n)$ takes a local maximum,
$\mathbf{B}_{increase}$ $h(x_{n+1}) - h(x_n) > valdelta$,
$\mathbf{B}_{ordinary}$ $h(x_n) > val$,

where $valmax$, $valdelta$, and val are arbitrary thresholds.

7 Large-Scale Experiments

7.1 Definition of Precision and Recall

Usually, when precision and recall are addressed in the Chinese word segmentation domain, they are calculated based on the number of *word*s. For example, consider a correctly segmented sequence "aaa|bbb|ccc|ddd", with a,b,c,d being characters and "|" indicating a word boundary. Suppose that the machine's result is "aaabbb|ccc|ddd"; then the correct words are only "ccc" and "ddd", giving a value of 2. Therefore, the precision is 2 divided by the number of words in the results (i.e., 3 for the words "aaabbb", "ccc", "ddd"), giving 67%, and the recall is 2 divided by the total number of words in the golden standard (i.e., 4 for the words "aaa","bbb", "ccc", "ddd") giving 50%. We call these values the word precision and recall, respectively, throughout this paper.

In our case, we use slightly different measures for the *boundary* precision and recall, which are based on the correct number of *boundaries*. These scores are also utilized especially in previous works on unsupervised segmentation (Ando and Lee, 2000) (Sun et al., 1998). Precisely,

$$Precision = \frac{N_{correct}}{N_{test}} \quad (6)$$

$$Recall = \frac{N_{correct}}{N_{true}}, \quad where \quad (7)$$

$N_{correct}$ is the number of correct boundaries in the result,

N_{test} is the number of boundaries in the test result, and,

N_{true} is the number of boundaries in the golden standard.

For example, in the case of the machine result being "aaabbb|ccc|ddd", the precision is 100% and the recall is 75%. Thus, we consider there to be no imprecise result as a boundary in the output of "aaabbb|ccc|ddd".

The crucial reason for using the boundary precision and recall is that boundary detection and word extraction are not exactly the same task. In this sense, assumption (A) or (B) is a general assumption about a *boundary* (of a sentence, phrase, word, morpheme). Therefore, the boundary precision and recall measure serves for directly measuring boundaries.

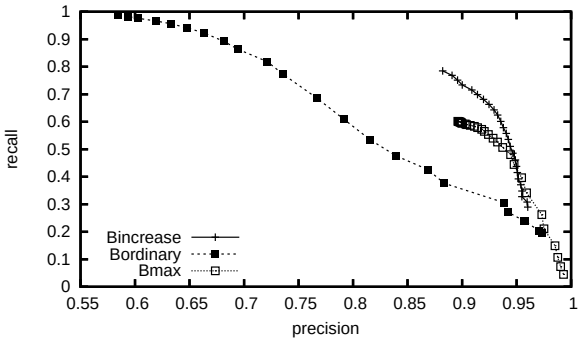

Figure 5: Precision and recall

Note that all precision and recall scores from now on in this paper are boundary precision and recall. Even in comparing the supervised methods with our unsupervised method later, the precision and recall values are all *recalculated* as boundary precision and recall.

7.2 Precision and Recall

The precision and recall graph is shown in Figure 5. The horizontal axis is the precision and the vertical axis is the recall. The three lines from right to left (top to bottom) correspond to $B_{increase}$ ($0.0 \leq valdelta \leq 2.4$), B_{max} ($4.0 \leq valmax \leq 6.2$), and $B_{ordinary}$ ($4.0 \leq val \leq 6.2$). All are plotted with an interval of 0.1. For every condition, the larger the threshold, the higher the precision and the lower the recall.

We can see how $B_{increase}$ and B_{max} keep high precision as compared with $B_{ordinary}$. We also can see that the boundary can be more easily detected if it is judged as comprising the proximity value of $h(x_n)$.

For $B_{increase}$, in particular, when $valdelta = 0.0$, the precision and recall are still at 0.88 and 0.79, respectively. Upon increasing the threshold to $valdelta = 2.4$, the precision is higher than 0.96 at the cost of a low recall of 0.29. As for B_{max}, we also observe a similar tendency but with low recall due to the smaller number of local maximum points as compared with the number of increasing points. Thus, we see how $B_{increase}$ attains a better performance among the three conditions. This shows the correctness of assumption (B).

From now on, we consider only $B_{increase}$ and proceed through our other experiments.

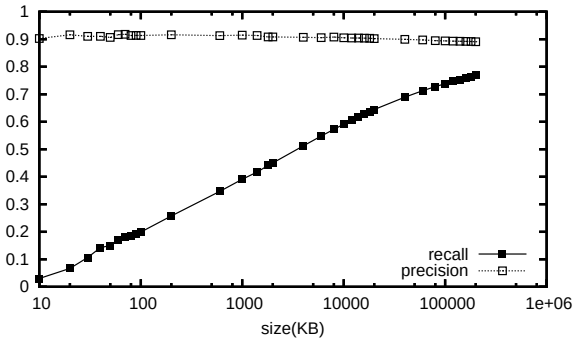

Figure 6: Precision and recall depending on training data size

Next, we investigated how the training data size affects the precision and recall. This time, the horizontal axis is the amount of learning data, varying from 10 KB up to 200 MB, on a log scale. The vertical axis shows the precision and recall. The boundary condition is $B_{increase}$ with $valdelta = 0.1$.

We can see how the precision always remains high, whereas the recall depends on the amount of data. The precision is stable at an amazingly high value, even when the branching entropy is obtained from a very small corpus of 10 KB. Also, the linear increase in the recall suggests that if we had more than 200 MB of data, we would expect to have an even higher recall. As the horizontal axis is in a log scale, however, we would have to have gigabytes of data to achieve the last several percent of recall.

7.3 Error Analysis

According to our manual error analysis, the top-most three errors were the following:

- Numbers: dates, years, quantities (example: 1998, written in Chinese number characters)
- One-character words (example:在(at) 又(again) 向(toward) 和(and))
- Compound Chinese words (example: 解放思想(open mind) being segmented into 解放(open) and 思想(mind))

The reason for the bad results with numbers is probably because the branching entropy for digits is less biased than for usual ideograms. Also, for one-character words, our method is limited, as we explained in §3. Both of these two problems, however, can be solved by applying special preprocessing for numbers and one-character words, given that many of the one-character words are functional characters, which are limited in number. Such improvements remain for our future work.

The third error type, in fact, is one that could be judged as correct segmentation. In the case of "open mind", it was not segmented into two words in the golden standard; therefore, our result was judged as incorrect. This could, however, be judged as correct.

The structures of Chinese words and phrases are very similar, and there are no clear criteria for distinguishing between a word and a phrase. The unsupervised method determines the structure and segments words and phrases into smaller pieces. Manual recalculation of the accuracy comprising such cases also remains for our future work.

8 Conclusion

We have reported an unsupervised Chinese segmentation method based on the branching entropy. This method is based on an assumption that "if the entropy of successive tokens increases, the location is at the context border." The entropies of n-grams were learned from an unsegmented 200-MB corpus, and the actual segmentation was conducted directly according to the above assumption, on 1 MB of test data. We found that the precision was as high as 90% with recall being around 80%. We also found an amazing tendency for the precision to always remain high, regardless of the size of the learning data.

There are two important considerations for our future work. The first is to figure out how to combine the supervised and unsupervised methods. In particular, as the performance of the supervised methods could be insufficient for data that are not from newspapers, there is the possibility of combining the supervised and unsupervised methods to achieve a higher accuracy for general data. The second future work is to verify our basic assumption in other languages. In particular, we should undertake experimental studies in languages written with phonogram characters.

References

R.K. Ando and L. Lee. 2000. Mostly-unsupervised statistical segmentation of Japanese: Applications to kanji. In *ANLP-NAACL*.

T.C. Bell, J.G. Cleary, and Witten. I.H. 1990. *Text Compression*. Prentice Hall.

Center for Chinese Linguistics. 2006. Chinese corpus. visited 2006, searchable from `http://ccl.pku.edu.cn/YuLiao_Contents.Asp`, part of it freely available from `http://www.icl.pku.edu.cn`.

T. Emerson. 2005. The second international chinese word segmentation bakeoff. In *SIGHAN*.

H.D. Feng, K. Chen, C.Y. Kit, and Deng. X.T. 2004. Unsupervised segmentation of chinese corpus using accessor variety. In *IJCNLP*, pages 255–261.

S.Z. Harris. 1955. From phoneme to morpheme. *Language*, pages 190–222.

A. Kempe. 1999. Experiments in unsupervised entropy-based corpus segmentation. In *Workshop of EACL in Computational Natural Language Learning*, pages 7–13.

J.K. Low, H.T Ng, and W. Guo. 2005. A maximum entropy approach to chinese word segmentation. In *SIGHAN*.

M. Sun, D. Shen, and B. K. Tsou. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *COLING-ACL*.

K. Tanaka-Ishii. 2005. Entropy as an indicator of context boundaries —an experiment using a web search engine —. In *IJCNLP*, pages 93–105.

H.P. Zhang, Yu H.Y., Xiong D.Y., and Q Liu. 2003. Hhmm-based chinese lexical analyzer ictclas. In *SIGHAN*. visited 2006, available from `http://www.nlp.org.cn`.