Chinese Part-of-Speech Tagging: One-at-a-Time or All-at-Once? Word-Based or Character-Based?

Hwee Tou Ng and **Jin Kiat Low**
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
{nght, lowjinki}@comp.nus.edu.sg

Abstract

Chinese part-of-speech (POS) tagging assigns one POS tag to each word in a Chinese sentence. However, since words are not demarcated in a Chinese sentence, Chinese POS tagging requires word segmentation as a prerequisite. We could perform Chinese POS tagging strictly after word segmentation (one-at-a-time approach), or perform both word segmentation and POS tagging in a combined, single step simultaneously (all-at-once approach). Also, we could choose to assign POS tags on a word-by-word basis, making use of word features in the surrounding context (word-based), or on a character-by-character basis with character features (character-based). This paper presents an in-depth study on such issues of processing architecture and feature representation for Chinese POS tagging, within a maximum entropy framework. We found that while the all-at-once, character-based approach is the best, the one-at-a-time, character-based approach is a worthwhile compromise, performing only slightly worse in terms of accuracy, but taking shorter time to train and run. As part of our investigation, we also built a state-of-the-art Chinese word segmenter, which outperforms the best SIGHAN 2003 word segmenters in the closed track on 3 out of 4 test corpora.

1 Introduction

Most corpus-based language processing research has focused on the English language. Theoretically, we should be able to just port corpus-based, machine learning techniques across different languages since the techniques are largely language independent. However, in practice, the special characteristics of different languages introduce complications. For Chinese in particular, words are not demarcated in a Chinese sentence. As such, we need to perform word segmentation before we can proceed with other tasks such as part-of-speech (POS) tagging and parsing, since one POS tag is assigned to each Chinese word (i.e., all characters in a Chinese word have the same POS tag), and the leaves of a parse tree for a Chinese sentence are words.

To build a Chinese POS tagger, the following questions naturally arise:

(1) Should we perform Chinese POS tagging strictly after word segmentation in two separate phases (one-at-a-time approach), or perform both word segmentation and POS tagging in a combined, single step simultaneously (all-at-once approach)?

(2) Should we assign POS tags on a word-by-word basis (like in English), making use of word features in the surrounding context (word-based), or on a character-by-character basis with character features (character-based)?

This paper presents an in-depth study on such issues of processing architecture and feature representation for Chinese POS tagging, within a maximum entropy framework. We analyze the performance of the different approaches in our attempt to find the best approach. To our knowledge, our work is the first to systematically investigate such issues in Chinese POS tagging.

2 Word Segmentation

As a first step in our investigation, we built a Chinese word segmenter capable of performing word segmentation without using POS tag information. Since errors in word segmentation will propagate to the subsequent POS tagging phase in the one-at-a-time approach, in order for our study to give relevant findings, it is important

that the word segmenter we use gives state-of-the-art accuracy.

The word segmenter we built is similar to the maximum entropy word segmenter of (Xue and Shen, 2003). Our word segmenter uses a maximum entropy framework and is trained on manually segmented sentences. It classifies each Chinese character given the features derived from its surrounding context. Each character can be assigned one of 4 possible boundary tags: "b" for a character that begins a word and is followed by another character, "m" for a character that occurs in the middle of a word, "e" for a character that ends a word, and "s" for a character that occurs as a single-character word.

2.1 Word Segmenter Features

Besides implementing a subset of the features described in (Xue and Shen, 2003), we also came up with three additional types of features ((d) − (f) below) which improved the accuracy of word segmentation. The default feature, boundary tag feature of the previous character, and boundary tag feature of the character two before the current character used in (Xue and Shen, 2003) were dropped from our word segmenter, as they did not improve word segmentation accuracy in our experiments.

In the following feature templates used in our word segmenter, C refers to a Chinese character while W refers to a Chinese word. Templates (a) − (c) refer to a context of five characters (the current character and two characters to its left and right). C_0 denotes the current character, C_n (C_{-n}) denotes the character n positions to the right (left) of the current character.

(a) $C_n\ (n=-2,-1,0,1,2)$
(b) $C_nC_{n+1}(n=-2,-1,0,1)$
(c) $C_{-1}C_1$
(d) W_0C_0
(e) $Pu(C_0)$
(f) $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$

For example, given the character sequence "新华社 记者", when considering the character "社", template (a) results in the following features C_{-2}=新 C_{-1}=华 C_0=社 C_1=记 C_2=者 to be set to 1, template (b) results in the features $C_{-2}C_{-1}$=新华 $C_{-1}C_0$=华社 C_0C_1=社记 C_1C_2=记者 to be set to 1.

2.2 Our Additional Features

W_0C_0 : This feature captures the word context in which the current character is found. For example, the character "社" within the word "新华社" will have the feature W_0C_0 =新华社_社 set to 1. This feature helps in recognizing seen words.

$Pu(C_0)$: A punctuation symbol is usually a good indication of a word boundary. This feature checks whether the current character is a punctuation symbol (such as "。", "-", " , ").

$T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$: This feature is especially helpful in predicting the word segmentation of dates and numbers, whose exact characters may not have been seen in the training text. Four type classes are defined: numbers represent class 1, dates ("日", "月", "年", the Chinese character for "day", "month", "year", respectively) represent class 2, English letters represent class 3, and other characters represent class 4. For example, when considering the character "年" in the character sequence "九〇年代R", the feature $T(C_{-2})\ldots T(C_2)$=11243 will be set to 1 ("九" and "〇" are the Chinese characters for "9" and "0" respectively).

2.3 Testing

During testing, the probability of a boundary tag sequence assignment $t_1...t_n$ given a character sequence $c_1...c_n$ is determined by using the maximum entropy classifier to compute the probability that a boundary tag t_i is assigned to each individual character c_i. If we were to just assign each character the boundary tag with the highest probability, it is possible that the classifier produces a sequence of invalid tags (e.g., "m" followed by "s"). To eliminate such possibilities, we implemented a dynamic programming algorithm which considers only valid boundary tag sequences given an input character sequence. At each character position i, the algorithm considers each last word candidate

ending at position *i* and consisting of *K* characters in length (*K* = 1, …, 20 in our experiments). To determine the boundary tag assignment to the last word *W* with *K* characters, the first character of *W* is assigned boundary tag "b", the last character of *W* is assigned tag "e", and the intervening characters are assigned tag "m". (If *W* is a single-character word, then the single character is assigned "s".) In this way, the dynamic programming algorithm only considers valid tag sequences, and we are also able to make use of the W_0C_0 feature during testing.

After word segmentation is done by the maximum entropy classifier, a post-processing step is applied to correct inconsistently segmented words made up of 3 or more characters. A word *W* is defined to be inconsistently segmented if the concatenation of 2 to 6 consecutive words elsewhere in the segmented output document matches *W*. In the post-processing step, the segmentation of the characters of these consecutive words is changed so that they are segmented as a single word. To illustrate, if the concatenation of 2 consecutive words "巴赛 罗纳" in the segmented output document matches another word "巴赛罗纳", then "巴赛 罗纳" will be re-segmented as "巴赛罗纳".

2.4 Word Segmenter Experimental Results

To evaluate the accuracy of our word segmenter, we carried out 10-fold cross validation (CV) on the 250K-word Penn Chinese Treebank (CTB) (Xia *et al.*, 2000) version 3.0. The Java opennlp maximum entropy package from sourceforge[1] was used in our implementation, and training was done with a feature cutoff of 2 and 100 iterations.

The accuracy of word segmentation is measured by recall (*R*), precision (*P*), and F-measure ($2RP/(R+P)$). Recall is the proportion of correctly segmented words in the gold-standard segmentation, and precision is the proportion of correctly segmented words in word segmenter's output.

Figure 1 gives the word segmentation F-measure of our word segmenter based on 10-fold CV on the 250K-word CTB. Our word segmenter achieves an average F-measure of 95.1%. This accuracy compares favorably with (Luo, 2003), which reported 94.6% word segmentation F-measure using his full parser without additional lexical features, and about 94.9%[2] word segmentation F-measure using only word boundaries information, no POS tags or constituent labels, but with lexical features derived from a 58K-entry word list.

The average training time taken to train on 90% of the 250K-word CTB was 12 minutes, while testing on 10% of CTB took about 1 minute. The running times reported in this paper were all obtained on an Intel Xeon 2.4GHz computer with 2GB RAM.

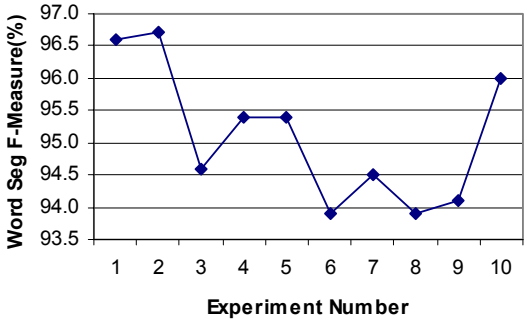

Figure 1: CTB 10-fold CV word segmentation F-measure for our word segmenter

As further evaluation, we tested our word segmenter on all the 4 test corpora (CTB, Academia Sinica (AS), Hong Kong CityU (HK), and Peking University (PK)) of the closed track of the 2003 ACL-SIGHAN-sponsored First International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003). For each of the 4 corpora, we trained our word segmenter on only the official released training data of that corpus. Training was conducted with feature cutoff of 2 and 100 iterations (these parameters were obtained by cross validation on the training set), except for the AS corpus where we used cutoff 3 since the AS training corpus was too big to train with cutoff 2.

Figure 2 shows our word segmenter's F-measure (based on the official word segmentation scorer of 2003 SIGHAN bakeoff) compared to those reported by all the 2003 SIGHAN participants in the four closed tracks (AS_c, HK_c, PK_c, CTB_c). Our word segmenter achieved higher F-measure than the best reported F-measure in the SIGHAN bakeoff on the AS_c, HK_c, and PK_c corpus. For CTB_c, due to the

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

[2] Based on visual inspection of Figure 3 of (Luo, 2003)

exceptionally high out-of-vocabulary (OOV) rate of the test data (18.1%), our word segmenter's F-measure ranked in the third position. (Note that the top participant of CTB_c (Zhang *et al.*, 2003) used additional named entity knowledge/data in their word segmenter).

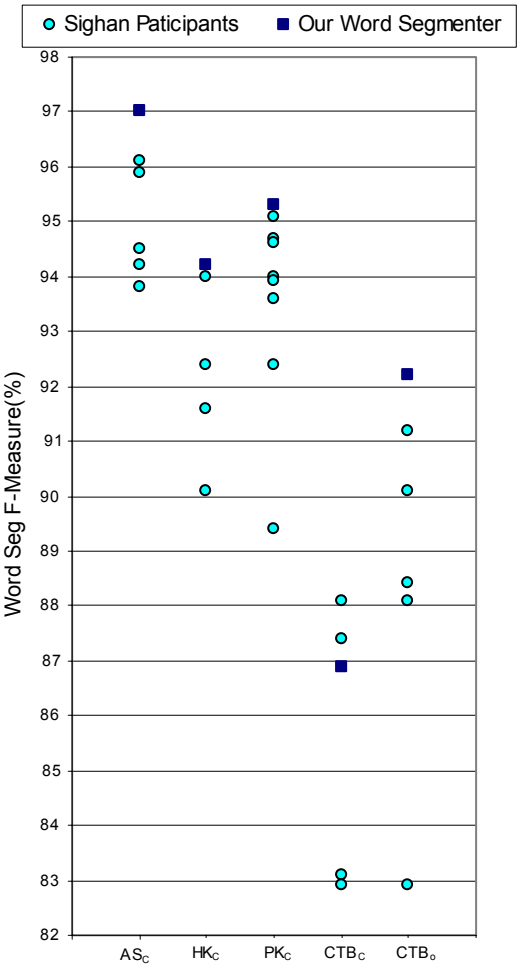

Figure 2: Comparison of word segmentation F-measure for SIGHAN bakeoff[3] tasks

We also compared the F-measure of our word segmenter on CTB_O, the open category of the CTB corpus, where participants were free to use any available resources and were not restricted to only the official released training data of CTB. On this CTB_O task, we used as additional training data the AS training corpus provided by SIGHAN, after converting the AS training corpus to GB encoding. We found that with this additional AS training data added to the original official released CTB training data of SIGHAN, our word segmenter achieved an F-measure of 92.2%, higher than the best reported F-measure in the CTB open task. With sufficient training data, our word segmenter can perform very well.

In our evaluation, we also found that the additional features we introduced in Section 2.2 and the post-processing step consistently improved average word segmentation F-measure, when evaluated on the 4 SIGHAN test corpora in the closed track. The additional features improved F-measure by an average of about 0.4%, and the post-processing step added on top of the use of all features further improved F-measure by 0.3% (i.e., for a cumulative total of 0.7% increase in F-measure).

3 One-at-a-Time, Word-Based POS Tagger

Now that we have successfully built a state-of-the-art Chinese word segmenter, we are ready to explore issues of processing architecture and feature representation for Chinese POS tagging.

An English POS tagger based on maximum entropy modeling was built by (Ratnaparkhi, 1996). As a first attempt, we investigated whether simply porting the method used by (Ratnaparkhi, 1996) for English POS tagging would work equally well for Chinese. Applying it in the context of Chinese POS tagging, Ratnaparkhi's method assumes that words are pre-segmented, and it assigns POS tags on a word-by-word basis, making use of word features in the surrounding context. This gives rise to a one-at-a-time, word-based POS tagger.

Note that in a one-at-a-time approach, the word-segmented input sentence given to the POS tagger may contain word segmentation errors, which can lower the POS tagging accuracy.

3.1 Features

The following feature templates were chosen. W refers to a word while POS refers to the POS tag assigned. The feature $Pu(W_0)$ checks if all characters in the current word are punctuation characters. Feature (e) encodes the class of characters that constitute the surrounding words (similar to feature (f) of the word segmenter in Section 2.1). Four type classes are defined: a word is of class 1 if it is a number; class 2 if the word is made up of only numeric characters followed by "日", "月", or "年"; class 3 when the word is made up of only English characters

[3] Last ranked participant of SIGHAN CTB (closed) with F-measure 73.2% is not shown in Figure 2 due to space constraint.

and optionally punctuation characters; class 4 otherwise.

(a) $W_n (n = -2,-1,0,1,2)$
(b) $W_n W_{n+1} (n = -2,-1,0,1)$
(c) $W_{-1} W_1$
(d) $Pu(W_0)$
(e) $T(W_{-2})T(W_{-1})T(W_0)T(W_1)T(W_2)$
(f) $POS(W_{-1})$
(g) $POS(W_{-2})POS(W_{-1})$

3.2 Testing

The testing procedure is similar to the beam search algorithm of (Ratnaparkhi, 1996), which tags each word one by one and maintains, as it sees a new word, the *N* most probable POS tag sequence candidates up to that point in the sentence. For our experiment, we have chosen *N* to be 3.

3.3 Experimental Results

The 250K-word CTB corpus, tagged with 32 different POS tags (such as "NR", "PU", etc) was employed in our evaluation of POS taggers in this study. We ran 10-fold CV on the CTB corpus, using our word segmenter's output for each of the 10 runs as the input sentences to the POS tagger. POS tagging accuracy is simply calculated as (number of characters assigned correct POS tag) / (total number of characters).

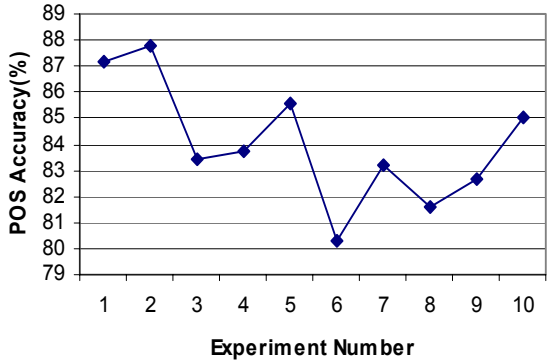

Figure 3: POS tagging accuracy using one-at-a-time, word-based POS tagger

The POS tagging accuracy is plotted in Figure 3. The average POS tagging accuracy achieved for the 10 experiments was only 84.1%, far lower than the 96% achievable by English POS taggers on the English Penn Treebank tag set. The average training time was 25 minutes, while testing took about 20 seconds. As an experiment, we also conducted POS tagging using only the features (a), (f), and (g) in Section 3.1, similar to (Ratnaparkhi, 1996), and we obtained an average POS tagging accuracy of 83.1% for that set of features.

The features that worked well for English POS tagging did not seem to apply to Chinese in the maximum entropy framework. Language differences between Chinese and English have no doubt made the direct porting of an English POS tagging method to Chinese ineffective.

4 One-at-a-Time, Character-Based POS Tagger

Since one-at-a-time, word-based POS tagging did not yield good accuracy, we proceeded to investigate other combinations of processing architecture and feature representation. We observed that character features were successfully used to build our word segmenter and that of (Xue and Shen, 2003). Similarly, character features were used to build a maximum entropy Chinese parser by (Luo, 2003), where his parser could perform word segmentation, POS tagging, and parsing in an integrated, unified approach. We hypothesized that assigning POS tags on a character-by-character basis, making use of character features in the surrounding context may yield good accuracy. So we next investigate such a one-at-a-time, character-based POS tagger.

4.1 Features

The features that were used for our word segmenter ((a) − (f)) in Section 2.1 were yet again applied, with two additional features (g) and (h) to aid POS tag prediction.

(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) $W_0 C_0$
(e) $Pu(C_0)$
(f) $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$
(g) $POS(C_{-1W_0})$
(h) $POS(C_{-2W_0})POS(C_{-1W_0})$

POS*(C_{-1W_0})*: This feature refers to the POS tag of the previous character before the current word. For example, in the character sequence "对 此 意见", when considering the

character “见”, the feature $POS(C_{-1W_0})$=PN is set to 1 (assuming “此” was tagged as PN).

$POS(C_{-2W_0})POS(C_{-1W_0})$: For the same example given above, when considering the character “见”, the feature $POS(C_{-2W_0})POS(C_{-1W_0})$=P_PN is set to 1 (assuming “对” was tagged as P and “此” was tagged as PN).

4.2 Testing

The testing algorithm is similar to that described in Section 3.2, except that the probability of a word being assigned a POS tag *t* is estimated by the product of the probability of its individual characters being assigned the same POS tag *t*. For example, when estimating the probability of “新华社” being tagged NR, we find the product of the probability of “新” being tagged NR, “华” being tagged NR, and “社” being tagged NR. That is, we enforce the constraint that all characters within a segmented word in the pre-segmented input sentence must have the same POS tag.

4.3 Experimental Results

10-fold CV for CTB is repeated for this POS tagger. Figure 4 shows the detailed POS tagging accuracy. With a one-at-a-time, character-based POS tagger, the average POS tagging accuracy improved to 91.7%, 7.6% higher than that achieved by the one-at-a-time, word-based POS tagger. The average training timing was 55 minutes, while testing took about 50 seconds.

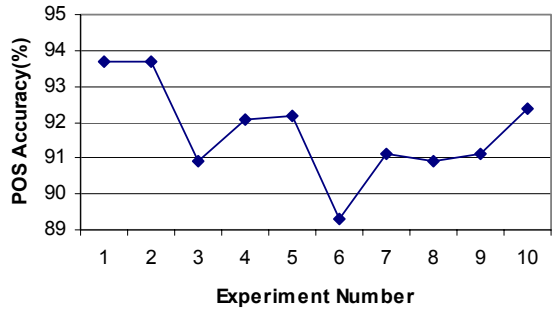

Figure 4: POS tagging accuracy using one-at-a-time, character-based POS tagger

When a paired t-test was carried out to compare character-based and word-based one-at-a-time approaches, the character-based approach was found to be significantly better than the word-based approach, at the level of significance 0.01.

Assuming a one-at-a-time processing architecture, Chinese POS tagging using a character-based approach gives higher accuracy compared to a word-based approach.

5 All-at-Once, Character-Based POS Tagger and Segmenter

Encouraged by the success of character features, we next explored whether a change in processing architecture, from one-at-a-time to all-at-once, while still retaining the use of character features, could give further improvement to POS tagging accuracy. In this approach, both word segmentation and POS tagging will be performed in a combined, single step simultaneously. Each character is assigned both a boundary tag and a POS tag, for example “b_NN” (i.e., the first character in a word with POS tag NN). Thus, given 4 possible boundary tags and 32 unique POS tags present in the training corpus, each character can potentially be assigned one of (4×32) classes.

5.1 Features

The features we used are identical to those employed in the character-based POS tagger described in section 4.1, except that features (g) and (h) are replaced with those listed below. In the following templates, *B* refers to the boundary tag assigned. For example, given the character sequence “对 此 意见”, when considering the character “见”, template (g) results in the feature $B(C_{-1W_0})POS(C_{-1W_0})$=s_PN to be set to 1. (assuming “此” was tagged as PN).

(g) $B(C_{-1W_0})POS(C_{-1W_0})$

(h) $B(C_{-2W_0})POS(C_{-2W_0})B(C_{-1W_0})POS(C_{-1W_0})$

Note that this approach is essentially that used by (Luo, 2003), since his parser performs both word segmentation and POS tagging (as well as parsing) in one unified approach. The features we used are similar to his tag features, except that we did not use features with three consecutive characters, since we found that the use of these features did not improve accuracy. We also added additional features (d) − (f).

5.2 Testing

Beam search algorithm is used with $N = 3$ during the testing phase.

5.3 Experimental Results

10-fold CV on CTB was carried out again, using unsegmented test sentences as input to the program.

Figure 5 shows the word segmentation F-measure, while Figure 6 shows the POS tagging accuracy achieved by this approach. With an all-at-once, character-based approach, an average word segmentation F-measure of 95.2% and an average POS tagging accuracy of 91.9% was achieved. The average training timing was 3 hours, while testing took about 20 minutes.

There is a slight improvement in word segmentation and POS tagging accuracy using this approach, compared to the one-at-a-time, character-based approach. When a paired t-test was carried out at the level of significance 0.01, the all-at-once approach was found to be significantly better than the one-at-a-time approach for POS tagging accuracy, although the difference was insignificant for word segmentation.

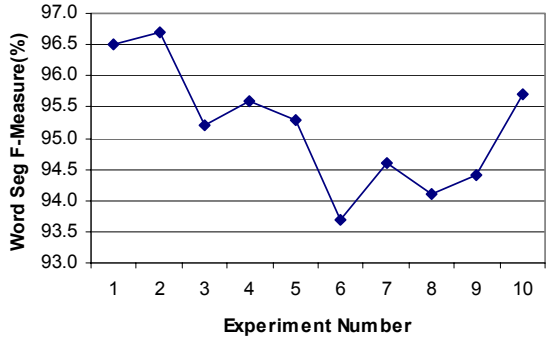

Figure 5: CTB 10-fold CV word segmentation F-measure using an all-at-once approach

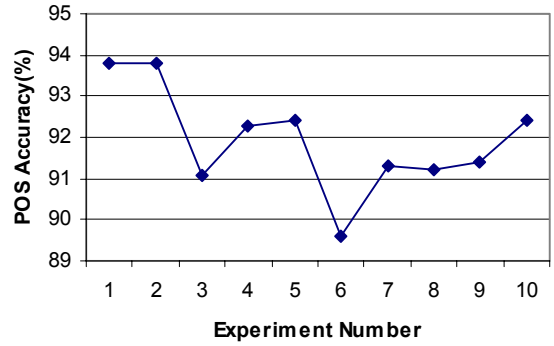

Figure 6: CTB 10-fold CV POS tagging accuracy using an all-at-once approach

However, the time required for training and testing is increased significantly for the all-at-once approach. When efficiency is a major consideration, or if high quality hand-segmented text is available, the one-at-a-time, character-based approach could indeed be a worthwhile compromise, performing only slightly worse than the all-at-once approach. Table 1 summarizes the methods investigated in this paper. Total testing time includes both word segmentation and POS tagging on 10% of CTB data. Note that an all-at-once, word-based approach is *not* applicable as word segmentation requires character features to determine the word boundaries.

Method	Word Seg F-measure (%)	POS Accuracy (%)	Total Testing Time
One-at-a-Time Word-Based	95.1	84.1	1 min 20 secs
One-at-a-Time Char-Based	95.1	91.7	1 min 50 secs
All-At-Once Char-Based	95.2	91.9	20 mins

Table 1: Summary table on the various methods investigated for POS tagging

6 Discussions

Word-based or character-based? The findings that a character-based approach is better than a word-based approach for Chinese POS tagging is not too surprising. Unlike in English where each English letter by itself does not possess any meaning, many Chinese characters have well defined meanings. For example, the single Chinese character "知" means "know". And when a character appears as part of a word, the word derives part of its meaning from the component characters. For example, "知识" means "knowledge", "无知" means "ignorant", "知名" means "well-known", etc. In addition, since the out-of-vocabulary (OOV) rate for Chinese words is much higher than the OOV rate for Chinese characters, in the presence of an unknown word, using the component characters in the word to help predict the correct POS tag is a good heuristic.

One-at-a-time or all-at-once? The all-at-once approach, which considers all aspects of available information in an integrated, unified

framework, can make better informed decisions, but incurs a higher computational cost.

7 Related Work

Much previous research on Chinese language processing focused on word segmentation (Sproat *et al.*, 1996; Teahan *et al.*, 2000; Sproat and Emerson, 2003). Relatively less work has been done on Chinese POS tagging. Kwong and Tsou (2003) discussed the implications of POS ambiguity in Chinese and the possible approaches to tackle this problem when tagging a corpus for NLP tasks. Zhou and Su (2003) investigated an approach to build a Chinese analyzer that integrated word segmentation, POS tagging and parsing, based on a hidden Markov model. Jing *et al.* (2003) focused on Chinese named entity recognition, considering issues like character-based versus word-based approaches. To our knowledge, our work is the first to systematically investigate issues of processing architecture and feature representation for Chinese POS tagging.

Our maximum entropy word segmenter is similar to that of (Xue and Shen, 2003), but the additional features we used and the post-processing step gave improved word segmentation accuracy.

The research most similar to ours is (Luo, 2003). Luo presented a maximum entropy character-based parser, which as a consequence of parsing also performed word segmentation and POS tagging. The all-at-once, character-based approach reported in this paper is essentially the approach proposed by Luo. While our investigation reveals that such an approach gives good accuracy, our findings however indicate that a one-at-a-time, character-based approach to POS tagging gave quite comparable accuracy, with the benefit of incurring much reduced computational cost.

8 Conclusion

Language differences between English and Chinese have made direct porting of an English POS tagging method to Chinese ineffective. In Chinese, individual characters encode information that aids in POS tagging. Using a character-based approach for Chinese POS tagging is more effective than a word-based approach. Our study has also revealed that the one-at-a-time, character-based approach gives relatively good POS tagging accuracy with a much improved training and testing time, compared with the all-at-once, character-based approach previously proposed.

9 Acknowledgements

This research is partially supported by a research grant R252-000-125-112 from National University of Singapore Academic Research Fund.

References

H. Jing, R. Florian, X. Luo, T. Zhang, and A. Ittycheriah. 2003. HowtogetaChineseName (Entity): segmentation and combination issues. In *Proc. of EMNLP*.

O. Y. Kwong and B. K. Tsou. 2003. Categorial fluidity in Chinese and its implications for part-of-speech tagging. In *Proc. of EACL*.

X. Luo. 2003. A maximum entropy Chinese character-based parser. In *Proc. of EMNLP*.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*.

R. Sproat, C. Shih, W. Gale, and N. Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377-404.

R. Sproat and T. Emerson. 2003. The first international Chinese word segmentation bakeoff. In *Proc. of SIGHAN Workshop*.

W. J. Teahan, Y. Wen, R. McNab, and I. H. Witten. 2000. A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, 26(3): 375-393.

F. Xia, M. Palmer, N. Xue, M. E. Okurowski, J. Kovarik, F-D Chiou, S. Huang, T. Kroch, and M. Marcus. 2000. Developing guidelines and ensuring consistency for Chinese text annotation. In *Proc. of LREC*.

N. Xue and L. Shen. 2003. Chinese word segmentation as LMR tagging. In *Proc. of SIGHAN Workshop*.

H-P Zhang, H-K Yu, D-Y Xiong, and Q. Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proc. of SIGHAN Workshop*.

G. Zhou and J. Su, 2003. A Chinese efficient analyser integrating word segmentation, part-of-speech tagging, partial parsing and full parsing. In *Proc. of SIGHAN Workshop*.