A Practical Morphological Analyzer Based on Penn Chinese Treebank Standard

Chooi-Ling Goh Yuchang Cheng Jia Lü Masayuki Asahara Yuji Mastumoto
Graduate School of Information Science, Nara Institute of Science and Technology
{ling-g,yuchan-c,jia-l,masayu-a,matsu}@is.naist.jp

1 Introduction

Till date, a freely usable Chinese morphological analysis system is still not widely available. Furthermore, since there is no single segmentation standard for all tagged corpora provided by different institution, some systems are available which are developed by the corpora providers, according to their segmentation standard. As far as we know, there is still no system (freely) available for Penn Chinese Treebank[1] (hereafter CTB) standard. Since this treebank is widely used by a lot of researches that do parsing, probably it is a good idea to build a practical morphological analyzer for CTB standard. The initial system that we built for CTB contains only 33,438 entries in the system dictionary. To build a practical system, this number is far from realization. Therefore, we tried to enlarge the system dictionary using unknown word extraction methods. We intend to extract a large amount of unknown words from a huge raw text corpus. Based on our methods, we have successfully increased the system dictionary to 120,769 entries. Although this number is good enough for a practical system but we still hope to add more in the future.

2 Two-layer Analysis

The architecture of the system has been described in [5]. First, we use a Hidden Markov Model-based analyzer to output the minimal unit segmentation and POS tagging (first layer). Then, a Support Vector Machine-based chunker is used to produce the output with CTB unit segmentation (second layer).

The definition of minimal unit has also been explained in [5]. Basically it defines the detailed groups for proper names and splits the numeral type and alpabetical type words into smaller units. The definition of proper names is for enriching the information captured in the segmentation output, and splitting numbers and foreign words is to ease the analysis process.

In order to build a practical system, we need a dictionary with reasonable size. We can retrieve the words from the training corpus but yet the size is too small when talking about real world text. In this paper, we describe some methods that we have used to enlarge the size of our dictionary. For evaluation purpose, we have used CTB version 4.0 (about 437,000 words) as our training corpus. The exclusive parts from version 5.0 (about 110,000 words) are used as testing data. Our basic dictionary is build from training data only.

[1] http://www.cis.upenn.edu/~chinese/

3 Preparation of System Dictionary

3.1 Extraction from CTB

We have made the changes to CTB 5.0 according to our segmentation unit. After that we extracted all words from CTB 4.0 to build our initial dictionary. We leave the exclusive part in CTB 5.0 as testing data (for evaluation). We have also removed some noise which we found not suitable to be used as the entries in the dictionary. Finally, we built an initial dictionary that contains 33,438 entries (word/POS pairs, a word can have more than one POS tag). There are 28,390 words if we consider the words only. To build a practical system, this number is far from realization. Therefore, we must find some ways to increase the number of entries in the dictionary.

3.2 Collection from Web

We have also made some collection from the web. These include place names (5,365 place names in China), country and capital names (391), and Chinese family names (436). These names are quite common on the web and they can be used directly in our system.

3.3 Unknown Word Extraction from Chinese Gigaword

Chinese Gigaword (CGW) is a raw text corpus provided by LDC. The size is about 1,118,380K Chinese characters. We use this corpus to extract new words to add into our system dictionary.

3.3.1 General Unknown Word Extraction and POS Tag Guessing

The unknown word extraction method used is similar to [4]. In this approach, we assign each character

with a character type such as NUMber, ALPphabet, SYMbol or HANzi, and label each character with BIES tagset[2]. We use Maximum Entropy models (hereafter ME) for the character-based tagging. We found that this method gives us the best unknown word recall although the precision is a bit lower. In [4], some pruning steps have been applied to delete some false unknown words. However, since this step deteriorates the recall, we do not do the pruning here as our purpose is to collect as many unknown words as possible.

Our evaluation is carried out using the test data in CTB 5.0. With the initial dictionary, there are about 9.2% of unknown word/POS pairs in the test data. Out of this, 7.8% are unknown words, 1.3% are unknown POS (the words exist in the dictionary but are with different POS tags). Currently, our method solves only the problem of unknown word but not the unknown POS. We leave the unknown POS problem for the future work.

The features that we use for tagging using ME are: 2 characters each from left and right contexts, character types, and 2 previously tagged labels. We get a recall of 72.2% for tokens, 72.1% for types, and 50.6% precision for types. In another words, we can get a quite high recall but the precision is not so good. Only about half of the words extracted are correct. However, since we want to increase the size of the dictionary, higher recall also means that we will get more words.

After unknown word extraction, we need to assign POS tags to them. We use the same method as described in [6] but we have expanded the feature set. We also apply ME model as the classification model. The training data are those words that appear only once in the corpus. This covers all major unknown POS tags. The features used are the context (unigram and bigram) and the internal component features (first character, last character and word length). The context features (known words and POS tags) are taken from the morphological analysis during the first layer analysis. Our experimental results show that the accuracy of tagging unknown words is 68.2% if only context features are used and 75.2% if all features are used.

Using this method, we tried on a small part of the CGW corpus for testing purpose. We estimated that 74.9% of the words extracted from CGW corpus are usable in our dictionary [5]. In a real run of the method to CGW, we extracted 51,412 word/POS pairs from file xin200209. Then we hired 4 native Chinese to check on the words manually in one month time. 14,537 words are correct, 10,643 words have been corrected with their POS tags. Since it is done manually, we also ask the checkers to correct some of the word boundaries to extract correct words (7,785 words). Finally, manually checking on the words give us a total of 26,281 (51.12%) correct words[3]. Although the result is a bit lower than our estimation, we still manage to get quite a number of new words.

[2]B - begin, I - inside, E - end, S - single

[3]There are some overlappings among these groups, so the final total is not the same as the total of all groups.

3.3.2 Person Name Extraction

In [3], we have seen that one can get better results if the extraction is focused on a certain type of unknown words, such as personal names. This is because we can train the system to be more precise to the type by providing specific features to it. For example, a Chinese person name normally comprises of a family name and a given name. A family name is normally one character long (very few with two characters) and it is almost a closed set. If we can provide the information about the family names, then it will be easier to guess the given names. At our disposal, we also have a set of characters that are possible to be used in transliteration foreign names. These provide some extra features for extraction.

The method that we use here is similar to the one described in [3]. First, an HMM-based analyzer is used to segment and POS tag the text, then an SVM-based chunker is used to extract the person names. Since our target is the Chinese given names and foreign names, we create a dictionary which consists of none of the both. It will make the HMM analyzer to wrongly segment all the names. In the second step, names are extracted by chunking process using SVM. We also provide family names and transliteration characters as the features. We assign each character with one of these 4 tags, FAM (family name), FOR (transliteration character), BTH (can be used for both), OTH (not in used for both). Currently we have collected 482 family names and 581 transliteration characters to be used for the training features. The context window is two characters at each left and right sides.

We have conducted an experiment using the CTB 5.0 test data. In CTB 4.0 there exist 4,190 given name and 926 foreign name instances. We use these data for training. In the test data, there are 1,157 given names and 194 foreign names. Table 1 shows the results of our method. Although we could get quite good accuracy with CJK given names, we could not get a good result with foreign names. This may be because the training data for the foreign names is not enough.

	Rec.	Prec.	F-mea.
CJK given name	89.02	70.12	78.45
Foreign name	39.69	56.62	46.67
Average	81.94	68.97	74.90

Table 1: Results for Person Name Extraction

Using this method, we extracted 4,622 person names from CGW, file xin199101. After manual checking, we obtained 3,976 (86%) words which are usable to our system. Since it is done manually, we also asked the checkers to correct some of the wrong POS and reassign boundaries if necessary. The accuracy for given names and foreign names only is

about 66%, follows our estimation during the testing experiments.

3.3.3 Checking with other Resources

From our past exprience, we realize that manual checking on unknown words in a time consuming task. Therefore, we also look for other solutions to speed up the process. One way is to use other resources for double checking as described below.

Sinica corpus[4] is the first tagged balanced corpus which contains about 5 millions words. Texts are collected from different areas and classified according to five criteria: genre, style, mode, topic, and source. Therefore, this corpus is a representative sample of modern Chinese language. Moreover, the size is 10 times larger than CTB.

Sinica corpus uses a different POS tagset as CTB corpus. It has 46 simplified POS tags, as compared to 33 tags in CTB. Basically the segmentation standard between CTB and Sinica is very similar but there are also some differences. From Sinica corpus, we could get around 150,000 distincts words. Despite the copyright problem to use the resources from Sinica, we cannot use the list of words directly from Sinica in our system since the segmentation standard is different. Therefore, we choose to use it in another way. First, we extract the new words from CGW using our unknown word extraction model. Instead of manual checking, we double check the words with Sinica corpus entries. If the words are found, then we assume that these words are correct ones. Since using a corpus requires copyright clearance, we have obtained the permission from the Academia Sinica to use their corpus as a reference.

In order to do this, first we need to compare the POS tagsets to find out equivalent POS tags. Table 2 shows the equivalent POS tags that we plan to use for comparison. We omitted some POS tags that cannot be matched directly, such as proper names, numbers, time nouns etc. As a results, we obtained a list of 105,030 word/POS pairs for comparison. We have applied the unknown word extraction model in Section 3.3.1 to the whole CGW corpus. We manage to extract 33,286 new entries which we are sure to be correct ones since they also exist in Sinica corpus.

We also manage to download a list of Chinese names from the web[5]. They provide a list of family names and a list of given names together with their frequency. From a total of 217,913 uniq names, they were able to give 619 distinct family names and 75,581 distinct given names. We found out that there are quite a lot of noise in the files because the way they cut the unique names into family name given names are not so reliable. Therefore, we decided not to use the family name list since we already have quite a number of them. However, we also do not want to use the given name list directly because it might contain error names as well. Our approach is the same as using Sinica corpus as a reference.

[4]http://www.sinica.edu.tw/SinicaCorpus
[5]http://www.geocities.com/hao520/namefreq

POS Tag	Sinica Tag	CTB Tag
Adjective	A	JJ
Adverb	D, Da, Dfa, Dfb, Dk	AD
Common noun	Na	NN
Localizer	Ncd	LC
Measure noun	Nf	M
Verb	V?[?], (+nom)	VV, NN
Stative verb	VH[?], (+nom)	VA, NN

Table 2: Matching between Sinica and CTB POS tags

First we extract the given names from the CGW using the method as described in Section 3.3.2, then we double check with the provided given name list to see if the names are inside the list. If they are in the list, then we assume that they are correct given names. By this way, we manage to extract 18,818 given names from CGW automatically.

4 Analysis Results

4.1 Minimal Unit Analysis - ChaSen

We use *ChaSen* [1] in our first layer analysis. Although *ChaSen* is originally built for Japanese language, it can be adopted easily to Chinese with slight modification. In fact, it is easier to setup the system in Chinese as we do not need to define grammar in Chinese since it does not have morphological changes such as inflection. We just need a training corpus and a dictionary for the training.

Top part of Table 3 shows the results of the first layer analysis. [CTB4 Dic] contains only the entries extracted from CTB 4.0, which is 33,438 entries. During the first phase of manual extraction from CGW and collection from the web, we mange to increase the dictionary to 68,626 entries [+ manual extraction]. At the second phase of extraction, with auto-checking with other resources, we further increase the dictionary to 120,769 entries [+ auto extraction]. We can see the improvement on the analysis results with the increment of size of dictionary. We realize a decreasing in unknown word rate. The row [no unknown] shows the results by retrieving all the entries from both training and testing data for building the dictionary. There are 39,896 entries in total, 6,458 entries more than [CTB4 Dic]. The training of HMM takes only the training data and the dictionary into account. The row [closed] shows the results where the training of HMM also includes the testing data. We can say that the [closed] is the perfect case of the system. From the results, we can see that our system is still far from perfect. Besides increasing the entries in the dictionary, we must also find a better way to improve the accuracy of POS tagging.

	Dictionary	Unknown rate	Segmentation			POS Tagging		
			Rec	Prec	F-mea	Rec	Prec	F-mea
First Layer	CTB4 Dic	9.2%	90.0	83.1	86.4	82.1	75.8	78.8
	+ manual extraction	7.4%	91.3	86.3	88.8	83.3	78.7	80.9
	+ auto extraction	5.4%	92.8	90.0	91.4	84.7	82.2	83.5
	no unknown	0%	97.1	97.8	97.4	90.1	90.7	90.4
	closed	0%	97.3	98.1	97.7	91.1	91.8	91.5
Second Layer	CTB4 Dic	-	88.5	81.1	84.6	80.2	73.6	76.7
	+ manual extraction	-	89.8	84.8	87.2	81.4	76.8	79.1
	+ auto extraction	-	91.4	88.8	90.1	83.0	80.6	81.8

Table 3: Results of First and Second Layer Analysis

4.2 CTB Unit Analysis - YamCha

The second layer is simple. It just take the output from the first layer and join the words by chunking. In order to obtain the original segmentation and POS tags, our task is to join up family names and given names, numbers, numeral type time nouns, and foreign words. The only difference with the original POS tags is that we cannot get back the original POS tags for foreign words. We used *YamCha* [7] for chunking as it is proved to be efficient for this task.

Bottom part of Table 3 shows the results of the second layer analysis. Compared to the results from the first layer analysis, the difference is quite small. This also means that the accuracy for chunking is high (about 76.7/78.8*100 = 97%) since the upper bound of the second layer depends on the accuracy of the first layer. By this way, we can easily convert the minimal unit segmentation back to CTB standard.

4.3 Related Work

There are some systems which are downloadable from the web. ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System)[6] [8] is an integrated system that uses an approach based on multi-layer Hidden Markov Models. ICTCLAS provides word segmentation, POS tagging and unknown word recognition. Their experiment results show that ICTCLAS achieved 98.25% accuracy for word segmentation, 95.63% for POS tagging with 24 tags and 93.38% with 48 tags. Their system is trained on Peking University corpus.

Microsoft Research Asia (MSR) also provides a free segmenter for download (S-MSRSeg)[7] [2]. It is a simplified model and does not provide the functionalities of new word identification, morphology analysis and adaptation to various standards. They applied a source-channel approach to word segmentation, and a class-based model and context model for new word identification. They obtained 95.5–96.2% recall and 95.0–95.6% precision for word segmentation and 60.4–78.4% recall and 46.2–68.1% precision for new word identification. MSR also defines their own segmentation standard.

[6] http://www.ict.ac.cn/freeware/003_ictclas.asp

[7] http://131.107.65.76/research/downloads/default.aspx

5 Conclusion

As a conclusion, a dictionary is very important in Chinese morphological analysis. The accuracy is worse if we have a small size dictionary. Our purpose is to build a practical system, therefore we look for some ways to enlarge the dictionary. We have increased the entries of our dictionary from 33,438 entries to 120,769 entries. However, we still wish to add more in the future as the accuracy of the system is still not near to the perfect.

References

[1] Matsumoto et. al, 2002. *Morphological Analysis System ChaSen 2.2.9 Manual.* Nara Institute of Science and Technology. http://chasen.naist.jp/.

[2] Jianfeng Gao, Andi Wu, Mu Li, Chang-Ning Huang, Hongqiao Li, Xinsong Xia, and Haowei Qin. 2004. Adaptive Chinese Word Segmentation. In *Proceedings of ACL*, pages 463–470.

[3] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2003. Chinese Unknown Word Identification Using Character-based Tagging and Chunking. In *Companion Volumn to the Proceedings of ACL 2003, Interactive Poster/Demo Sessions*, pages 197–200.

[4] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2004. Pruning False Unknown Words to Improve Chinese Word Segmentation. In *Proceedings of PACLIC 18*, pages 139–149.

[5] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2005. The Development of Chinese ChaSen. In *Proceedings of Gengo-shori-gakkai Nenzi Taikai Ronbun-shu.* (In Japanese)

[6] Chooi-Ling Goh. 2003. Chinese Unknown Word Identification by Combining Statistical Models. Master's thesis, Nara Institute of Science and Technology, Japan.

[7] Taku Kudo and Yuji Matsumoto. 2001. Chunking with Support Vector Machines. In *Proceedings of NAACL*, pages 192–199.

[8] Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. HHMM-based Chinese Lexical Analyzer ICTCLAS. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 184–187.