Chinese Tagging Based on Maximum Entropy Model

Ka Seng Leong
Faculty of Science and Technology of University of Macau
Av. Padre Tomás Pereira, Taipa, Macau, China
ma56538@umac.mo

Fai Wong
Faculty of Science and Technology of University of Macau, INESC Macau
Av. Padre Tomás Pereira, Taipa, Macau, China
derekfw@umac.mo

Yiping Li
Faculty of Science and Technology of University of Macau
Av. Padre Tomás Pereira, Taipa, Macau, China
ypli@umac.mo

Ming Chui Dong
Faculty of Science and Technology of University of Macau, INESC Macau
Av. Padre Tomás Pereira, Taipa, Macau, China
dmc@inesc-macau.org.mo

Abstract

In the Fourth SIGHAN Bakeoff, we took part in the closed tracks of the word segmentation, part of speech (POS) tagging and named entity recognition (NER) tasks. Particularly, we evaluated our word segmentation model on all the corpora, namely Academia Sinica (CKIP), City University of Hong Kong (CITYU), University of Colorado (CTB), State Language Commission of P.R.C. (NCC) and Shanxi University (SXU). For POS tagging and NER tasks, our models were evaluated on CITYU corpus only. Our models for the evaulation are based on the maximum entropy approach, we concentrated on the word segmentation task for the bakeoff and our best official results on all the corpora for this task are 0.9083 F-score on CITYU, 0.8985 on CKIP, 0.9077 on CTB, 0.8995 on NCC and 0.9146 on SXU.

1 Introduction

In the Fourth SIGHAN Bakeoff, besides providing the evaluation tasks for the word segmentation and NER, it also introduced another important evaluation task, POS tagging for Chinese language. In this bakeoff, our models built for the tasks are similar to that in the work of Ng and Low (2004). The models are based on a maximum entropy framework (Ratnaparkhi, 1996; Xue and Shen, 2003). They are trained on the corpora for the tasks from the bakeoff. To understand the model, the implementation of the models is wholly done ourselves. We used Visual Studio .NET 2003 and C++ as the implementation language. The Improved Iterative Scaling (IIS) (Pietra et al., 1997) is used as the parameter estimation algorithm for the models. We tried all the closed track tests of the word segmentation, the CITYU closed track tests for POS tagging and NER.

2 Maximum Entropy

In this bakeoff, our basic model is based on the framework described in the work of Ratnaparkhi (1996) which was applied for English POS tagging. The conditional probability model of the framework is called maximum entropy (Jaynes, 1957). Maximum entropy model is a feature-based, probability model which can include arbitrary number of features that other generative models like N-gram model, hidden Markov model (HMM) (Rabiner, 1989) cannot do. The probability model can be defined over $X \times Y$, where X is the set of

possible histories and *Y* is the set of allowable futures or classes. The conditional probability of the model of a history *x* and a class *y* is defined as

$$p_\lambda(y \mid x) = \frac{\prod_i \lambda_i^{f_i(x,y)}}{Z_\lambda(x)} \quad (1)$$

$$Z_\lambda(x) = \sum_y \prod_i \lambda_i^{f_i(x,y)} \quad (2)$$

where λ is a parameter which acts as a weight for the feature in the particular history. The equation (1) states that the conditional probability of the class given the history is the product of the weightings of all features which are active under the consideration of (*x*, *y*) pair, normalized over the sum of the products of all the classes. The normalization constant is determined by the requirement that $\sum_y p_\lambda(y \mid x) = 1$ for all *x*.

To find the optimized parameters λ of the conditional probability is one of the important processes in building the model. This can be done through a training process. The parameter estimation algorithm used for training is Improved Iterative Scaling (IIS) (Pietra et al., 1997) in our case. In training the models for this bakeoff, the training data is given in the form of a sequence of characters (for the tasks of word segmentation and NER) or words (POS tagging) and their classes (tags), the parameters λ can be chosen to maximize the likelihood of the training data using *p*:

$$L(p) = \prod_{i=1}^{n} p_\lambda(x_i, y_i) = \prod_{i=1}^{n} \frac{1}{Z_\lambda(x)} \prod_{j=1}^{m} \lambda_j^{f_j(x_i, y_i)} \quad (3)$$

But of course, the success of the model depends heavily on the selection of features for a particular task. This will be described in Section 5.

3 Chinese Word Segmenter

We concentrated on the word segmentation task in this bakeoff. For the Chinese word segmenter, it is based on the work that treats Chinese word segmentation as tagging (Xue and Shen, 2003; Ng and Low, 2004). Given a Chinese sentence, it assigns a so-called boundary tag to each Chinese character in the sentence. There are four possible boundary tags: *S* for a character which is a single-character word, *B* for a character that is the first character of a multi-character word, *E* for a character that is the last character of a multi-character word and *M* for a character that is neither the first nor last of a multi-character word. With these boundary tags, the word segmentation becomes a tagging problem where each character in Chinese sentences is given one of the boundary tags which is the most probable one according to the conditional probability calculated by the model. And then sequences of characters are converted into sequences of words according to the tags.

4 POS Tagger and Named Entity Recognizer

For the POS tagging task, the tagger is built based on the work of Ratnaparkhi (1996) which was applied for English POS tagging. Because of the time limitation, we could only try to port our implemented maximum entropy model to this POS tagging task by using the similar feature set (discussed in Section 5) for a word-based POS tagger as in the work of Ng and Low (2004). By the way, besides porting the model to the POS tagging task, it was even tried in the NER task by using the same feature set (discussed in Section 5) as used for the word segmentation in order to test the performance of the implemented model.

The tagging algorithm for these two tasks is basically the same as used in word segmentation. Given a word or a character, the model will try to assign the most probable POS or NE tag for the word or character respectively.

5 Features

To achieve a successful model for any task by using the maximum entropy model, an important step is to select a set of useful features for the task. In the following, the feature sets used in the tasks of the bakeoff are discussed.

5.1 Word Segmentation Features

The feature set used in this task is discussed in our previous work (Leong et al., 2007) which is currently the best in our implemented model. They are the unigram features: C_{-2}, C_{-1}, C_0, C_1 and C_2, bigram features: $C_{-2}C_{-1}$, $C_{-1}C_0$, C_0C_1, C_1C_2 and $C_{-1}C_1$ where C_0 is the current character, C_n (C_{-n}) is the

character at the n^{th} position to the right (left) of the current character. For example, given the character sequence "維多利亞港" (Victoria Harbour), while taking the character "利" as C_0, then C_{-2} = "維", $C_{-1}C_1$ = "多亞", etc. The boundary tag (*S*, *B*, *M* or *E*) feature T_{-1} is also applied, i.e., the boundary tag assigned to the previous character of C_0. And the last feature WC_0: This feature captures the word context in which the current character is found. It has the format "W_C_0". For example, the character "利" is a character of the word "維多利亞港". Then this will give the feature WC_0 = "維多利亞港_利".

5.2 POS Tagging Features

For this task, because of the time limitation as mentioned in the previous section, we could only port our implemented model by using a part of the feature set which was used in the word-based tagger discussed in the work of Ng and Low (2004). The feature set includes: W_n (n = -2 to 2), W_nW_{n+1} (n = -2, -1, 0, 1), $W_{-1}W_1$, $POS(W_{-2})$, $POS(W_{-1})$, $POS(W_{-2})POS(W_{-1})$ where *W* refers to a word, *POS* refers to the POS assigned to the word and *n* refers to the position of the current word being considered. For example, while considering this sentence taken from the POS tagged corpus of CITYU: "香港/Ng 特別/Ac 行政區/Nc 正式/Dc 成立/Vt" (Hong Kong S.A.R. is established), taking "行政區" as W_0, then W_{-2} = "香港", $W_{-1}W_1$ = "特別 正式", $POS(W_{-2})$ = "Ng", $POS(W_{-2})POS(W_{-1})$ = "Ac Dc", etc.

5.3 Named Entity Recognition Features

For the NER task, we directly used the same feature set as for the word segmentation basically. However, because the original NE tagged corpus is presented in two-column format, where the first column consists of the character and the second is a tag, a transformation which is to transform the original corpus to a sentence per line format before collecting the features or other training data is needed. This transformation actually continues to read the lines from the original corpus, whenever a blank line is found, a sentence of characters with NE tags can be formed.

After that, the features collected are the unigram features: C_{-2}, C_{-1}, C_0, C_1 and C_2, bigram features: $C_{-2}C_{-1}$, $C_{-1}C_0$, C_0C_1, C_1C_2 and $C_{-1}C_1$, NE tag features: T_{-1}, WC_0 (this feature captures the NE context in which the current character is found) where T_{-1} refers to the NE tag assigned to the previous character of C_0, *W* refers to the named entity. So similar to the explanation of features of word segmentation, for example, given the sequence from the NER tagged corpus of CITYU: "一/N 個/N 中/B-LOC 國/I-LOC 人/N" (One Chinese), while taking the character "中" as C_0, then C_{-2} = "一", $C_{-1}C_1$ = "個國", WC_0 = "中國__中", etc.

For all the experiments conducted, training was done with a feature cutoff of 1.

6 Testing

For word segmentation task, during testing, given a character sequence $C_1 \ldots C_n$, the trained model will try to assign a boundary tag to each character in the sequence based on the probability of the boundary tag calculated. Then the sequence of characters is converted into sequence of words according to the tag sequence $t_1 \ldots t_n$. But if each character was just assigned the boundary tag with the highest probability, invalid boundary tag sequences would be produced and wrong word segmentation results would be obtained. In particular, known words that are in the dictionary of the training corpus are segmented wrongly because of these invalid tag sequences. In order to correct these, the invalid boundary tag sequences are collected, such as for two-character words, they are "B B", "B S", "M S", "E E", etc., for three-character words, they are "B E S", "B M S", etc., and for four-character words, they are "B M M S", "S M M E", etc. With these invalid boundary tag sequences, some post correction to the word segmentation result can be tried. That is after the model tagger has done the tagging for a Chinese sentence every time, the invalid boundary tag sequences will be searched within the preliminary result given by the tagger. When the invalid boundary tag sequence is found, the characters corresponding to that invalid boundary tag sequence will be obtained. After, the word formed by these characters is looked up to see if it is indeed a word in the dictionary, if it is, then the correction is carried out.

Another kind of post correction to the word segmentation result is to make some guessed correction for some invalid boundary tag sequences such as "B S", "S E", "B B", "E E", "B M S", etc. That is, whenever those tag sequences are met

within the preliminary result given by the model tagger, they will be corrected no matter if there is word in the dictionary formed by the characters corresponding to the invalid boundary tag sequence.

We believe that similar post correction can be applied to the NER task. For example, if such NE tag sequences "B-PER N", "N I-PER N", etc. occur in the result, then the characters corresponding to the invalid NE tag sequence can be obtained again and looked up in the named entity dictionary to see if they really form a named entity. However, we did not have enough time to adapt this for the NER task finally. Therefore, no such post correction was applied for the NER task in this bakeoff finally.

7 Evaluation Results

We evaluated our models in the closed tracks of the word segmentation, part of speech (POS) tagging and named entity recognition (NER) tasks. Particularly, our word segmentation model was evaluated on all the corpora, namely Academia Sinica (CKIP), City University of Hong Kong (CITYU), University of Colorado (CTB), State Language Commission of P.R.C. (NCC) and Shanxi University (SXU). For POS tagging and NER tasks, our models were evaluated on the CITYU corpus only. Table 1 shows our official results for the word segmentation task in the bakeoff. The columns R, P and F show the recall, precision and F-score respectively.

Run_ID	R	P	F
cityu_a	0.9221	0.8947	0.9082
cityu_b	0.9219	0.8951	0.9083
ckip_a	0.9076	0.8896	0.8985
ckip_b	0.9074	0.8897	0.8985
ctb_a	0.9078	0.9073	0.9075
ctb_b	0.9077	0.9078	0.9077
ncc_a	0.8997	0.8992	0.8995
ncc_b	0.8995	0.8992	0.8994
sxu_a	0.9186	0.9106	0.9145
sxu_b	0.9185	0.9107	0.9146

Table 1. Official Results in the Closed Tracks of the Word Segmentation Task on all Corpora

We submitted a few runs for each of the tests of the corpora. Table 1 shows the best two runs for each of the tests of the corpora for discussion here. The run (a) applied only the post correction to the known words that are in the dictionary of the training corpus but are segmented wrongly because of the invalid boundary tag sequences. The run (b) applied also the guessed post correction for some invalid boundary tag sequences in the results as mentioned in Section 6. From the results above, it can be seen that the runs with the guessed post correction generally gave a little bit better performance than those that did not apply. This shows that the guess somehow made some good guesses for some unknown words that appear in the testing corpora.

Table 2 shows our official results for the POS tagging task. The columns A shows the accuracy. The columns IV-R, OOV-R and MT-R show the recall on in-vocabulary words, out-of-vocabulary words and multi-POS words (multi-POS words are the words in the training corpus and have more than one POS-tag in either the training corpus or testing corpus) respectively. The run (a) used the paramters set which was observed to be the optimal ones for the model in the training phase. The run (b) used the parameters set of the model in the last iteration of the training phase.

Run_ID	A	IV-R	OOV-R	MT-R
cityu_a	0.1890	0.2031	0.0550	0.1704
cityu_b	0.2793	0.2969	0.1051	0.2538

Table 2. Official Results in the Closed Track of the POS Tagging Task on the CITYU Corpus

It can be seen that our results were unexpectedly low in accuracy. After releasing the results, we found that the problem was due to the encoding problem of our submitted result files. The problem probably occurred after the conversion from our Big5 encoded results to the UTF-16 encoded results which are required by the bakeoff. Therefore, we did the evaluation ourselves by running our POS tagger again, using the official evaluation program and the truth test set. Finally, our best result was 0.7436 in terms of accuracy but this was still far lower than the baseline (0.8425) of the CITYU corpus. This shows that the direct porting of English word-based POS tagging to Chinese is not effective.

Table 3 shows our official results for the NER task. The columns R, P and F show the recall, precision and F-score respectively. Again, similar to the POS tagging task, the run (a) used the

paramters set which was observed to be the optimal ones for the model in the training phase. The run (b) used the parameters set of the model in the last iteration of the training phase.

Run_ID	R	P	F
cityu_a	0.0874	0.1058	0.0957
cityu_b	0.0211	0.0326	0.0256

Table 3. Official Results in the Closed Track of the NER Task on the CITYU Corpus

It can be seen that our results were again unexpectedly low in accuracy. The cause of such low accuracy results was due to parts of the wrong format of the submitted result files compared with the correct format of the result file. So like the POS tagging task, we did the evaluation ourselves by running our NE recognizer again. Finally, our best result was 0.5198 in terms of F-score but this was again far lower than the baseline (0.5955) of the CITYU corpus. This shows that the similar feature set for the word segmentation task is not effective for the NER task.

8 Conclusion

This paper reports the use of maximum entropy approach for implementing models for the three tasks in the Fourth SIGHAN Bakeoff and our results in the bakeoff. From the results, we got good experience and knew the weaknesses of our models. These help to improve the performance of our models in the future.

Acknowledgements

The research work reported in this paper was partially supported by "Fundo para o Desenvolvimento das Ciências e da Tecnologia" under grant 041/2005/A.

References

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging, in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Philadelphia, USA, pages 133-142.

Edwin Thompson Jaynes. 1957. Information Theory and Statistical Mechanics, *The Physical Review*, 106(4): 620-630.

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

Ka Seng Leong, Fai Wong, Yiping Li, and Ming Chui Dong. 2007. Chinese word boundaries detection based on maximum entropy model, in *Proceedings of the 11th International Conference on Enhancement and Promotion of Computational Methods in Engineering and Science (EPMESC-XI)*, Kyoto, Japan.

Lawrence Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2): 257–286.

Nianwen Xue, and Libin Shen. 2003. Chinese word segmentation as LMR tagging, in *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, Sapporo, Japan, pages 176-179.

Steven Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE transactions on pattern analysis and machine intelligence*, 19(4): 380–393.