Part-of-Speech Tagging with Both Character and Word Information

You Zhou[1, a], Fangzhou Liu[2, b]

[1]College of Mathematics and Statistics, Hunan University of Finance and Economics, Changsha, China

[2]College of Mathematics and Computer Science, Hunan Normal University, Changsha, China

[a]zhouyou84@163.com, [b]ark.new@163.com

Keywords: Part-of-speech tagging; Character information; Word information; Maximum entropy.

Abstract. Part-of-speech tagging is to determine an appropriate grammatical category for each word in a sentence, which is one of the basic tasks of natural language processing. The former part-of-speech tagging methods mostly study the co-occurrence probability of the adjacent parts of speech at the word level, and lack the analysis of the internal structure of the word. In this paper, we propose a maximum entropy based Chinese part-of-speech tagger which not only uses word and part-of-speech information, but also uses character information inside the word. Our approach gives an error reduction of 61.3%, compared to the approach using only the word information.

Introduction

Part-of-speech (POS) is the most frequently used shallow grammatical information. In various natural language processing tasks, including information retrieval, machine translation etc, POS plays a very important role. Some Chinese words have more than one POS. For example, the word "da3" in the phrase "da3 jiang4 you2" is a verb, but in "yi4 da3 ji1 dan4" "da3" becomes a quantifier. One of the main difficulties in POS tagging is to determine the appropriate POS of this kind of multi-category words. Although the proportion of multi-category words in the dictionary is not high, multi-category words appear frequently in the actual text, indicating that many common words are multi-category words. Another difficulty of POS tagging is to determine the POS of out-of-vocabulary (OOV) words.

With the vigorous development of large corpus in natural language processing, various data-driven approaches, such as transformation-based learning [1], hidden markov model [2], maximum entropy model [3], conditional random fields [4], have been investigated to solve the POS tagging problem. However, most of these methods only considered the word-level information such as word or POS collocation, and did not analyze the internal structure of words. In fact, Chinese words are composed of Chinese characters which usually have some sort of grammatical properties, so there are grammatical structures within the words. We can use the internal structure of words to identify their POS. For example, the character "len3" in the words "len3 xiao4" and "len3 cang2" has the adverb property, and the character "she4" in the words "fan3 she4" and "zhe2 she4" has the verb property. The OOV word "len3 she4" consists of "len3" and "she4", so it can be judged as a verb according to the structure of "adverb + verb". In this paper, we propose a maximum entropy based Chinese part-of-speech tagger which uses both character-level information and word-level information. Experimental results show that, the introduction of character features obviously improves the accuracy of Chinese POS tagging.

Maximum Entropy Framework

Maximum entropy (ME) model [5] has been widely used in natural language processing in recent years, since it allows the inclusion of diverse forms of information without causing data fragmentation and necessarily assuming independence between the predictors. The central idea of ME model is to seek the probability distribution with the maximum entropy, over the set of the ones

subject to certain constraints. Such constraints force the model to match its feature expectations with those observed in the training set. A feature of ME model is a binary-value function $f_i(x, y)$ as follows:

$$f_i(x, y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are both true} \\ 0 & \text{otherwise} \end{cases} . \quad (1)$$

In the scenario of POS tagging, x denotes the contextual information, and y denotes the POS tag. Given an event (x, y), the conditional probability $p(y \mid x)$ can be computed as:

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{i=1}^{k} \alpha_i^{f_i(x,y)} . \quad (2)$$

Where α_i is the weight of feature f_i, which can be estimated by GIS or IIS [5] algorithm, and $Z(x)$ is the normalization factor:

$$Z(x) = \sum_{y} \prod_{i=1}^{k} \alpha_i^{f_i(x,y)} . \quad (3)$$

Feature Selection

A trained ME model is composed of its features and their weights, so the performance of ME model depends largely on the effectiveness of the features, which is extracted from the training corpus according to the feature templates. Therefore, feature templates determine the value space of features, and have the greatest influence on the behavior of ME model.

Character Feature. According to our statistics on Chinese corpus, Chinese words are of length of about 1.7 characters on average. Therefore, we extract character information from a context window of 5 characters which roughly covers the context from the previous word to the next word. Table 1 lists all the character feature templates.

Table 1 Character feature templates

C_{-2}, C_{-1}, C_0, C_1, C_2
$C_{-2}C_{-1}$, $C_{-1}C_0$, C_0C_1, C_1C_2, $C_{-1}C_1$
$Pu(C_0)$
$T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$

Where C refers to a Chinese character and the subscript number indicates the offset from the current character C_0. For instance, "$C_{-1}C_1$" means the immediately previous character and the immediately succeeding character. $Pu(C_0)$ checks whether the current character is a punctuation symbol. T represents the character type. Four types are defined: Type 1 represents a number; Type 2 represents the date (year, month, day); Type 3 represents an English letter; Type 4 represents other characters do not belong to the former three types.

Word Feature. 4 kinds of word information in a context window of -2~+2 words are employed in this work. All the word feature templates are shown in Table 2.

Table 2 Word feature templates

W_{-2}, W_{-1}, W_0, W_1, W_2
$W_{-1}W_0$, W_0W_1, $W_{-1}W_1$
P_{-1}
$P_{-2}P_{-1}$
P_{0f1}, P_{0f2}

Where W and P denote the word and POS tag respectively, and the subscript number indicates the offset from the current word W_0. We also made statistics on the frequency of each POS of each word in the training corpus, and then use P_{0f1} and P_{0f2} to represent the most frequent and the second most frequent POS of the current word respectively. If the current word is an OOV word, the values of P_{0f1} and P_{0f2} are "OOV". When the current word has only one POS in the training corpus, P_{0f2} is set to "none".

After extracting feature instances from the training set according to the feature templates, it is also needed to filter the features in order to remove noise. There are usually two algorithms to select ME feature, CCFS (Count Cutoff Feature Selection) and IFS (Incremental Feature Selection) [6]. Ratnaparkhi [6] pointed out that, IFS algorithm had heavy computation complexity, and took much long time to train the model, while did not always outperform CCFS algorithm. In this paper, CCFS algorithm is applied to feature selection.

POS sequence search

Our POS tagger is based on characters, that is, the ME model is trained by character samples, and the output is the probability of a character being assigned a POS tag. The probability of a word being assigned a POS tag is estimated by the product of the probability of its individual characters being assigned the same POS tag. For example, when estimating the probability of "fang1 zhou1" being tagged NR (the person name tag), we find the product of the probability of "fang1" being tagged NR and "zhou1" being tagged NR.

Due to the need of the POS tags of the previous two words when estimating the probability of the POS tag of the current word, we embed the ME model into the beam search algorithm [3] to find the best POS sequence. Beam search algorithm simplifies the Viterbi algorithm to a local optimal algorithm, in order to reduce the time complexity. It 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.

Evaluation and Discussion

Our experimental corpus is derived from the word segmentation and POS tagging corpus [7] produced by the Institute of Computational Linguistics of Peking University. 6 months of People's Daily corpus (about 20 million characters) with 44 kinds of POS, which has been manually segmented word and assigned POS tag, is divided into the training set, development set and test set according to an 8:1:1 ratio. The cutoff value setting of ME model are implemented on the development set. POS tagging accuracy is simply calculated as (number of words assigned correct POS tag) / (total number of words).

Cutoff Value Setting. In order to determine the optimal cutoff value, we compare the performance of a series of cutoff values on the development set. The ME models are trained with the feature templates listed in both Table 1 and Table 2, and with 100 iterations. Table 3 shows the results. The best cutoff value is 2, that is, the features appear less than or equal to 2 times in the training set are thrown out.

Table 3 Comparison of cutoff values

Cutoff Value	POS Tagging Accuracy
0	94.0%
1	94.2%
2	95.2%
3	94.2%
4	94.7%
5	93.8%

It is observed that most of the discarded low-frequency features are composite features. Although these features have strong discriminability to identify POS tag, but they are not statistically stable enough, and easily lead to over-fitting, so it is helpful to delete them for improving the overall quality of feature set. However, if the cutoff value is set too high, a lot of useful information will be lost, thus reducing the performance of ME model. So it is necessary to choose a suitable cutoff value.

Feature Comparison. A performance comparison between character feature templates in Table 1 and word feature templates in Table 2 is made with a feature cutoff of 2 and 100 iterations. As shown in Table 4, the accuracy of character feature templates achieves 92.5%, 5.6% higher than that of word feature templates. Combining these two feature templates achieves the best performance, with an error reduction of 61.3% over the word feature templates, which verify our analysis that the character information inside a Chinese word is very helpful to identify its POS.

Table 4 Comparison of feature templates

Feature Templates	POS Tagging Accuracy
Character	92.5%
Word	87.6%
Character & Word	95.2%

Conclusion

In this paper, we present a ME based Chinese POS tagger with both character information and word information, which is much more effective than the traditional approach of using only the word information. Therefore, characters inside words do encode information that aids in POS tagging. Another advantage of the character-based approach is that it can accurately identify the POS of OOV words, and does not need to build a special POS tagging module for OOV words, which simplifies the design of POS tagger.

Acknowledgements

This work was supported by the Natural Science Foundation of Hunan Province (No. 13JJ6030 and No. 2015JJ2021), the Scientific Research Foundation of Higher Education Institutions of Hunan Province (No. 14C0192), and the Science and Technology Foundation of Changsha City (No. K1308031-11 and No. K1406014-11).

References

[1] E. Brill: Computational Linguistics Vol. 21 (1995), p. 543

[2] Q. Liu, H. P. Zhang and H. K. Yu: Journal of Computer Research and Development Vol. 41 (2004), p. 1421

[3] A. Ratnaparkhi: Proceedings of the Conference on Empirical Methods in Natural Language Processing (1996), p. 133

[4] J. D. Yu, Y. Q. Ge and Z. T. Yu: Microelectronics & Computer Vol. 28 (2011), p. 63

[5] A. L. Berger, S. A. D. Pietra and V. J. D. Pietra: Computational Linguistics Vol. 22 (1996), p. 39

[6] A. Ratnaparkhi: Ph.D. Dissertation, University of Pennsylvania (1998)

[7] http://icl.pku.edu.cn/icl_groups/corpus/dwldform1.asp