An Enhanced Model

for Chinese Word Segmentation and Part-of-speech Tagging

Jiang Feng

Department of Computer Science and Engineering Shanghai Jiao Tong University Shanghai, China, 200030 f_jiang@sjtu.edu.cn

Chen Yuquan

Department of Computer Science and Engineering Shanghai Jiao Tong University Shanghai, China, 200030 yqchen@mail.sjtu.edu.cn

Abstract

This paper will present an enhanced model for Chinese probabilistic word segmentation and part-of-speech (POS) tagging. The model introduces the information of Chinese word length as one of its features to reach a more accurate result. And in addition, the model also achieves the integration of segmentation and POS tagging. After presenting the model, this paper will give a brief discussion on how to solve the problems in statistics and how to further integrate Chinese Named Entity Recognition into the model. Finally, some figures of experiments and comparisons will be reported. which shows that the accuracy of word segmentation is 97.09%, and the accuracy of POS tagging is 98.77%.

1 Introduction

Generally, Chinese Lexical Analysis consists of two phases; one is word segmentation and the other is part-of-speech(POS) tagging. Rule-based approach and statistic-based approach are two dominant ways in natural language processing, as well as Chinese Lexical Analysis. This paper will only focus on the later one. Hence, our model is called a probabilistic model.

Scanning through the researches in this field before, we have just found two points at which the performance of a Chinese word segmentation and POS tagging system could get better. One is the on the system architecture, and the other is from the Machine Learning theory. Liu Hui Department of Computer Science and Engineering Shanghai Jiao Tong University Shanghai, China, 200030 Ih_Charles@sjtu.edu.cn

Lu Ruzhan

Department of Computer Science and Engineering Shanghai Jiao Tong University Shanghai, China, 200030 rzlu@mail.sjtu.edu.cn

First, the traditional way of Chinese Lexical Analysis simply regards the word segmentation and POS tagging as two separated phases. Each one of them has its own algorithms and models. Dividing the whole process into two independent parts can lower the complexity of the design of system, but decrease the performance as well, because the two are fully integrated when a human sentence. Fortunately, processing а manv researchers have already noticed it, and recent projects pay more attention on the integration of word segmentation and POS tagging, such as [Gao Shan, Zhang Yan. 2001]'s pseudo trigram integrated model, [Fu Guohong et al. 2001]'s analyzer which incorporates backward Dynamic Programming and A* algorithm, [Sun Maosong, et al. 2003]'s 'Divide and Conquer integration', [Zhang Huaping, et al. 2003]'s hierarchical hidden Markov model and so on. The experiments given by these papers also showed a great potential of the integrated modek.

Besides the system architecture, another point should be noticed. A probabilistic model of word segmentation and POS tagging can be regarded as an instance of Machine Learning. In Machine Learning, the feature extraction is the most important aspect, and far more important than a learning algorithm. In the models nowadays, it seems that the features for Chinese Lexical Analysis are a little too simple. Most of them take tag sequences, or word frequencies as the distinguishing features and ignore the other useful information that are provided by Chinese itself.

In this paper, we will present an enhanced, not too complex, model for word segmentation and POS tagging, which will not only inherit the merit of an integrated model, but also take a new feature (word length) into account.

The second part of this paper will describe the model, including the input, output, and some assumptions. The third part will give some brief discussion about the model on some issues like data sparseness and Named Entity Recognition. In the final part, the results of our experiments will be reported.

2 The Model

The first step to establish the model is to make a formal description for its input and output. Here, a Chinese word segmentation and POS tagging system is viewed as with input,

 $C_1, C_2, ..., C_n$

where C_i is the i'th Chinese character of the input sentence, and with output pairs, $(m \le n)$

 $\begin{pmatrix} L_1 \\ T_1 \end{pmatrix} \begin{pmatrix} L_2 \\ T_2 \end{pmatrix} \cdots \begin{pmatrix} L_m \\ T_m \end{pmatrix}$

where L_i is the word length of the i'th word in the segmented word sequence, T_i is the word tag, and each (L_i, T_i) pair is corresponding to a

segmented and tagged word, and
$$\sum_{i=1}^{n} i = n$$
.

It is easily seen that the distinction between this model and other models is that this one introduces word length. In fact, word length really works, and affects the performance of the system in a great deal, of which our later experiments will approve.

The motivation to introduce word length into our model is initially from the classical Chinese poems. When we read these poems, we may spontaneously obey some laws in where to have a pause. For example, in most cases, a 7-characterlined Jueju(A kind of poem format) is read as **/**/***. And the pauses in a sentence are much related to the length of words or chunks. Even in modern Chinese, word length also plays a part. Sometimes we prefer to use disyllabic words rather than single one, though both are correct in grammar. For example, in our daily lives, we always say "汽车/n 修理/v 站/n" or "汽车/n 维修/v站/n", but seldom hear "汽车/n修/v站/n", where '修理", "维修" and '修" have the same meaning. So, it is reasonable to assume that the occurrence of the word length will obey some unwritten laws when human writes or speaks. Introducing the word length into the word segmentation and POS tagging model may be in accord with the needs for processing Chinese.

Another main characteristic of the model is that it is an integrated model, because there is only one hop through the input sentence to the output wordtag sequence.

The following text will introduce how the model works. We will also inherit n-gram assumption in our model.

Our destination is to find a sequence of (L_i, T_i) pairs that maximizes the probability,

(For conveniece, we will use \vec{A} to represent a squence of $A_1, A_2, A_3...$)

And,

$$P((\overrightarrow{L,T}) | \overrightarrow{C}) = \frac{P(\overrightarrow{C} | (\overrightarrow{L,T})) * P(\overrightarrow{L,T})}{P(\overrightarrow{C})} \dots 2.2$$

For $P(\vec{C})$ is a constant given a \vec{C} , we just need to consider $P(\vec{C} \mid (\vec{L}, \vec{T}))$ and $P(\vec{L}, \vec{T})$.

First consider $P(\vec{C} \mid (\vec{L}, \vec{T}))$. Suppose \vec{W} is the vertex of words that (\vec{L}, \vec{T}) represents(i.e. the segmented word sequence), and the dependency assumption is like the following Bayers Network:

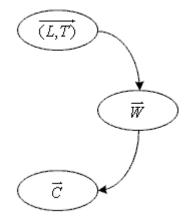


Figure 2.1: Dependency assumption among length-tag pair, word and character

So, we have,

 $P(\vec{C} \mid (\vec{L}, \vec{T})) = P(\vec{C} \mid \vec{W}) * P(\vec{W} \mid (\vec{L}, \vec{T})) \dots 2.3$ Because \vec{W} is the segmentation of \vec{C} , $P(\vec{C} \mid \vec{W})$ is always 1, and by another assumption that the occurrence of every word is independent to each other, then

where $P(W_i | (L_i, T_i))$ means the conditional probability of W_i under L_i and T_i . For example,

P("学习"|2, v) is the conditional probability of "学习" under a 2-charactered verb which may be computed as (the number of "学习" appearing as a verb) / (the number of all 2-charactered verbs). With 2.3 and 2.4, $P(\vec{C} \mid (\vec{L}, \vec{T}))$ is ready.

Then consider $P(\overrightarrow{L,T})$, which is easy to retrieve when we apply n-gram assumption. Suppose n is 2, which means that (L_i, T_i) only depends on (L_{i-1}, T_{i-1}) .

Here $P((L_i, T_i) | (L_{i-1}, T_{i-1}))$ means the probability of a Tag T_i with Length L_i appearing next to Tag T_{i-1} with Length L_{i-1} , which may be computed as (the number of $(L_{i-1}, T_{i-1})(L_i, T_i)$ appearing in corpus) / (the number of (L_{i-1}, T_{i-1}) appearing in corpus). So, $P(\overline{L,T})$ is also ready.

Combining formula 2.1, 2.2, 2.3, 2.4 and 2.5, we have,

Now, the enhanced model is complete with 2.6. When establishing the model, we have made several assumptions.

- 1. the dependency assumption between tag-length pairs, words and characters like the Bayers network of figure 2.1
- 2. Word and word are independent.
- 3. n-gram assumption on (*T*,*L*) pairs.

The validation of these assumptions is still somewhat in doubt, but the computational complexity of the model is decreased.

All the resources required to achieve this model are also listed, i.e., a word list with probability $P(W_i | \begin{pmatrix} L_i \\ T_i \end{pmatrix})$, and an n-gram transition network with probability. $p(L_i) | (L_{i-1}) | (L_{i-1})$

network with probability $P\left(\begin{pmatrix} L_i \\ T_i \end{pmatrix} | \begin{pmatrix} L_{i-n+1} \\ T_{i-n+1} \end{pmatrix} ..., \begin{pmatrix} L_{i-1} \\ T_{i-1} \end{pmatrix} \right)$

The algorithm to implement this model is also rather simple, and using Dynamic Programming, we could finish the algorithm in O(cn), where n is the length of input sentence, and c is a constant related to the maximum ambiguity in a position.

3 Discussion

Though the model itself is not difficult to implement as we have presented in last section, there are still some problems that we will be probably encountered with in practice. The first one is the data sparseness when we do the statistics. Another is how to further integrate Chinese Named Entity Recognition into the new, word-lengthintroduced model.

3.1 Data Sparseness

The Data Sparseness happens when we are calculating $P(\begin{pmatrix} L_i \\ T_i \end{pmatrix})\begin{pmatrix} L_{i-n+1} \\ T_{i-n+1} \end{pmatrix}, \dots, \begin{pmatrix} L_{i-1} \\ T_{i-1} \end{pmatrix}$. After the word length is introduced, the need for larger

corpus is greatly increased. Suppose we are using a tri-gram assumption on length-tag pairs, the number of tags is 28 as that of our system, and the max word length is 6, then the number of patterns we should count is,

28 * 6 * 28 * 6 * 28 * 6 = 4,741,632.

To retrieve a reasonable statistical result, the scale of the corpus should at least be several times larger than that value. It is common that we don't have such a large corpus, and meet the problem so called Data Sparseness.

One way to deal with the problem is to find a good smoothing, and another is to make further independent assumption between word length and word tag. The word length sequence and word tag sequence can be considered independent. That means,

Now, the patterns to count are just as many as those of a traditional n-gram assumption that only assumes the dependency among tags.

3.2 Named Entity Recognition Integration

Named Entity Recognition is one of the most important parts of word segmentation and POS tagging systems, for the words in word list are limited while the language seems infinite. There are always new words appearing in human language, among which human names, place names and organization names are most common and most valuble to recognize. The performance of Named Entity Recognition will have a deep impact on the performance of a whole word segmentation and POS tagging system. The research on Named Entity Recognition has appeared for many years. No matter whether the current performance of Named Entity Recognition is ideal or not, we will not discuss it here, and instead, we will just show how to integrate the existing Name Entity Recognition methods into the new model.

During the integration, more attention should be paid to the structural and probabilistic consistency. For structural consistency, the original system structure does not need modifying when a new method of Named Entity Recognition is applied. For probabilistic consistency, the probabilities outputted by the Named Entity Recognition should be compatible with the probabilities of the words in the original word list.

Here, we will take the Human Name Recognition as an example to show how to do the integration.

[Zheng Jiahen, et al. 2000] has presented a probabilistic method for Chinese Human Name Recognition, which is easy to understand and suitable to be borrowed as a demonstration.

That paper defined the probability for a Chinese Human Name as:

$$P(ns \mid ik) = F(i) * E(k) \dots 3.2$$

$$P(np \mid iik) = F(i) * M(i) * E(k) \dots 3.3$$

which may be a human name, "ns" means a single name when "j" is empty, "np" means plural name when "j" is not empty, F(i) is the probability of "i" being a family name, M(i) means the probability of "j" being the middle character of a human name, E(k) means the probability of "k" being the tailing character of a human name, $P(ns \mid ik)$ is the probability of "ik" being a single name, and $P(np \mid$ *ijk*) is the probability of *"ijk"* being a plural name. F(i), M(j), and E(k) are easily retrieved from corpus, so $P(ns \mid ik)$ and $P(np \mid ijk)$ can be known.

However, P(ns | ik) and P(np | ijk) do not satisfy the requirements of the word length introduced model. The model needs probabilities like P(w|(l,t)), where w is a word, t is a word tag, and *l* is the word length. Therefore, $P(ns \mid ik)$ needs to be modified into $P(ik \mid nh, 2)$, for *ik* is always a 2charactered word, and likewise, $P(np \mid ijk)$ needs to be modified into $P(ijk \mid nh, 3)$, where "nh" is the word tag for human name in our system.

 $P(ns \mid ik)$ is equivalent to $P(nh, 2 \mid ik)$ and $P(np \mid ik)$ *ijk*) is equivalent to $P(nh, 3 \mid ijk)$. $P(ns \mid ik)$ can be converted into $P(ik \mid nh, 2)$ through following way,

 $P(ik \mid nh, 2)$

$$=\frac{P(nh,2 | ik)P(ik)}{P(nh,2)} = \frac{P(nh,2 | ik)P(i)P(k)}{P(nh,2)}$$

where "i", "k" have the same meaning with those in 3.2 and 3.3. and *nh* is the tag for human name.

In this formula, 'i'' and 'k'' are assumed to be independent. P(nh, 2), P(i), P(k) are easy to retrieve, which represent the probability of a 2charactered human name, the probability of character "i" and the probability of character "k". $P(nh, 2 \mid ik)$ is computed from 3.2. Thus, the conversion of $P(nk \mid nh, 2)$ to $P(nh, 2 \mid ik)$ is done.

In the same way, $P(np \mid ijk)$ can be converted into $P(ijk \mid nh, 3)$ by: $P(ijk \mid nh,3)$

$$=\frac{P(nh,3 | ijk)P(ijk)}{P(nh,3)} = \frac{P(nh,3 | ik)P(i)P(j)P(k)}{P(nh,3)}$$

Finally, the Human Name Recognition Module is integrated into the whole system. The input string $C_1, C_2, ..., C_n$ first goes through the Human Name Recognition module, and the module outputs a temporary word list, which consists of a column of words that are probably human names and a column of probabilities corresponding to the words, which can be computed by 3.4 and 3.5. The whole system then merges the temporary word list and the original word list into a new word list, and applies the new word list in segmenting and tagging C_1, C_2, \ldots, C_n .

Conclusion & Experiments 4

This paper has presented an enhanced probabilistic model of Chinese Lexical Analysis, which introduces word length as one of the features and achieves the integration of word segmentation, Named Entity Recognition and POS tagging.

At last, we will briefly give the results of our experiments. In the previous experiments, we have compared many simple probabilistic models for Chinese word segmentation and POS tagging, and found that the system using maximum word frequency as segmentation strategy and forward tri-gram Markov model as POS tagging model (MWF + FTMM) reaches the best performance. Our comparisons will be done between the MWF+FTMM and the enhance model with trigram assumption. The training corpus is 40MB annotated Chinese text from People's Daily. The testing data is about 1MB in size and is from People's Daily, too.

	MWF+FTMM	New Model
WSA	95.24%	97.09%
PTA	97.12%	98.77%
Total	92.50%	95.90%

Table 4.1: The accuracy by word, with named entity not considered

	MWF+FTMM	New Model
WSA	93.86%	95.68%
РТА	93.89%	95.72%
Total	88.13%	91.59%

Table 4.2: The accuracy by word,with named entity considered

	MWF+FTMM	New Model
WSA	69.46%	82.63%
PTA	72.58%	80.33%
Total	50.42%	66.38%

Table 4.3: The accuracy by sentence, with named entity not considered

	MWF+FTMM	New Model
WSA	63.86%	74.78%
PTA	61.40%	67.41%
Total	39.21%	50.42%

Table 4.4: The accuracy by sentence,with named entity considered

NOTES:

- MWF: Maximum Word Frequency, a very simple strategy in word segmentation disambiguation, which chooses the word sequence with max probability as its result.
- FTMM: Forward Tri-gram Markov Model, a popular model in POS tagging.
- MWF+FTMM: A strategy, which chooses the output that makes a balance between the MWF and FTMM as its result.
- WSA (by word): Word Segmentation Accuracy, measured by recall, i.e. the number of correct segments divided by the number of segments in corpus.

(In a problem like word segmentation, the result of precision measurement is commonly around that of recall measurement.)

- PTA (by word): POS Tagging Accuracy based on correct segmentation, the number of words that are correctly segmented and tagged divided by the number of words that are correctly segmented.
- Total (by word): total accuracy of the system, measured by recall, i.e. the number of words that are correctly segmented and tagged divided by the number of words in corpus, or simply WSA * PTA.
- WSA (by sentence): the number of correctly segmented sentences divided by the number of sentences in corpus. A correctly segmented sentence is a sentence whose words are all correctly segmented.

PTA (by sentence): the number of correctly tagged sentences divided by the number of correctly segmented sentences in corpus. A correctly tagged sentence is a sentence whose words are all correctly segmented and tagged.

Total (by sentence): WSA * PTA.

Named entity considered or not: When named entity is not considered, all the unknown words in corpus are deleted before evaluation. Otherwise, nothing is done on the corpus.

According to the results above (Table 4.1, Table 4.2, Table 4.3, Table 4.4), the new enhanced model does better than the MWF + FTMM in every field. Introducing the word length into a Chinese word segmentation and POS tagging system seems effective.

This paper just focuses on the pure probabilistic model for word segmetation and POS tagging. It can be predicted that, with more disambiguation strategies, such as some rule based approaches, being implemented into the new model to achieve a multi-engine system, the performance will be further improved.

5 Acknowledgements

Thank Fang Hua and Kong Xianglong for their previous work, who have just graduated.

References

- Sun Maosong, Xu Dongliang, Benjamin K Tsou. 2003. Integrated Chinese word segmentation and part-of-speech tagging based on the divide-andconquer strategy. International Conference on Natural Language Processing and Knowledge Engineering Proceedings, Beijing.
- Zhang Huaping, Liu Qun, et al. 2003. *Chinese lexical analysis using hierarchical hidden Markov model.* 2nd SIGHAN workshop affiliated with 41th ACL, Sapporo Japan
- Fu Guohong, Wang Ping, Wang Xiaolong. 2001. Research on the approach of integrating chinese wordd segmentation with part-of-speech tagging. Application Research of Computer. (In Chinese)
- Gao Shan, Zhang Yan. 2001. The Research on Integrated Chinese Word Segmentation and Labeling based on trigram statistical model. Natural Language Understanding & Machine Translation (JSCL-2001), Taiyuan. (In Chinese)
- Zheng Jiahen, Li Xin, et al. 2000. *The Research of Chinese names recognition method based on corpus. Journal of Chinese Information Processing.* (In Chinese)