Context-rule Model for Pos Tagging

Yu-Fang Tsai
Academia Sinica, Institute of Information Science
128 Sec. 2Academy Rd.
Nankang, Taipei, Taiwan 115
eddie@iis.sinica.edu.tw

Keh-Jiann Chen
Academia Sinica, Institute of Information Science
128 Sec. 2Academy Rd.
Nankang, Taipei, Taiwan 115
kchen@iis.sinica.edu.tw

Abstract

Part-of-speech tagging for a large corpus is a labour intensive and time-consuming task. In order to achieve fast and high quality tagging, algorithms should be high precision and in particular, its tagging results should require less manual proofreading. In this paper, we proposed a context-rule model to achieve both the above goals for pos tagging.

We compared the tagging precisions between Markov bi-gram model and context-rule classifier. According to the experiments, context-rule classifier performs better than those two other algorithms. Also, it covers the data sparseness problem by utilizing more context features, and reduces the amount of corpus that is need to be manual proofread by introducing the confidence measure.

1 Introduction

Part-of-speech tagging for a large corpus is a labour intensive and time-consuming task. In order to achieve fast and high quality tagging, algorithms should be high precision and in particular, its tagging results should require less manual proofreading. There is lots of work on part-of-speech tagging such as Hidden Markov Models (HMMs), Maximum Entropy Models (MEs), and Support Vector Machines (SVMs), etc. Most of works addressed on the high accuracy of tagging results only. In this paper, we proposed a context-rule model to achieve both the above goals for pos tagging.

2 Tagging Algorithms

In this study, we are going to test two different tagging algorithms based on same training data and testing data. The two tagging algorithms are Markov bi-gram model, and context-rule classifier. For Markov bi-gram model, we propose a new form named word-dependent Markov bi-gram model, which will be described later. The training data and testing data are extracted from Sinica corpus, a 5 million word balanced Chinese corpus with pos tagging (Chen et al., 1996). The confidence measure will be defined for each algorithm and the best accuracy will be estimated at the constraint of only a fixed amount of testing data being proofread.

It is easier to proofread and make more consistent tagging results, if proofreading processes were done by checking the keyword-in-context file for each ambivalence word and only the tagging results of ambivalence word need to be proofread. The words with single category need not be rechecked their pos tagging. For instance, in Table 1, the keyword-in-context file of the word '研究' (research), which has pos-categories of verb type *VE* and noun type *Nv*, is sorted according to its left/right context. The proofreader can see the other examples as references to determine whether or not each tagging result is right. If all of the occurrences of ambivalence word have to be rechecked, it is still too much of the work.

The common terms used in the following tagging algorithms were defined as follows:

w_k	The k-th word in a sequence
c_k	The pos-category associated with k-th word w_k

的(DE)	重要(VH)	研究(**Nv**)	機構(Na)	之(DE)
相當(Dfa)	重視(VJ)	研究(**Nv**)	開發(Nv)	，(COMMACATEGORY)
民族(Na)	音樂(Na)	研究(**VE**)	者(Na)	明立國(Nb)
赴(VCL)	香港(Nc)	研究(**VE**)	該(Nes)	地(Na)
亦(D)	值得(VH)	研究(**VE**)	。(PERIODCATEGORY)	
合宜性(Na)	值得(VH)	研究(**VE**)	。(PERIODCATEGORY)	
更(D)	值得(VH)	研究(**Nv**)	。(PERIODCATEGORY)	

Table 1 Sample keyword-in-context file of the words '研究' sorted by its left/right context

$w_1c_1,...,w_nc_n$ A word sequence containing n words with their associated categories respectively

2.1 Markov Bi-gram Model

The most widely used tagging models are part-of-speech n-gram models, in particular bi-gram and tri-gram model. In a bi-gram model, it looks at pair of categories (or words) and uses the conditional probability of $P(c_k \mid c_{k-1})$, and the Markov assumption is that the probability of a category occurring depends only on the one category before it.

Given a word sequence $w_1,...w_n$, the Markov bi-gram model calculates the probability of each candidate category c_k for a target word w_k by $P(c_k \mid c_{k-1})\ P(w_k \mid c_k)$. There are two approaches to estimate the statistical data for $P(c_k \mid c_{k-1})$. One is to count all the occurrences in the training data, called general Markov model, and another one is to count only the occurrences in which each w_k occurs, called word-dependent Markov model. We compared the two different approaches of Markov bi-gram model with the proposed context-rule model algorithm in the experiments.

2.2 Context-rule Model

The conventional Markov n-gram models utilize the features of categories of context words and the probability distribution of the categories of target words. In fact, for some cases the best pos-tags might be determined by other context features, such as context words instead of the categories. In the context-rule model, we extend the scope of the dependency context of a target word into its 2 by 2 context windows. Therefore the context features of a target word w_0 can be represented by the vector of $[w_{-2}, c_{-2}, w_{-1}, c_{-1}, w_1, c_1, w_2, c_2]$. Each feature vector may be associated with one or more pos-tags. The association probability of the candidate category c_0' is $P(c_0' \mid w_0, \textit{feature vector})$. If for some (w_0, c_0'), the value of $P(c_0' \mid w_0, \textit{feature vector})$ is not 1, it means that the pos of w_0 cannot be uniquely determined by its context vector. Some additional features have to be incorporated to resolve the ambiguity. If for some word w_0, all of its pos c_0' such that the value of $P(c_0' \mid w_0, \textit{feature vector})$ is zero which means there is no training examples with the same context vector of w_0. If the full scope of the context feature vector is used, data sparseness problem will seriously hurt the system performance. Therefore partial feature vectors are used instead of full feature vectors. The partial feature vectors applied in our context-rule classifier are w_{-1}, w_1, $c_{-2}c_{-1}$, c_1c_2, $c_{-1}c_1$, $w_{-2}c_{-1}$, $w_{-1}c_{-1}$, and c_1w_2.

At the training stage, for each feature vector type many rule instances will be generated. For instance, with the above applied feature vector types, we can extract rule patterns of w_{-1}(先生), w_1(之餘), $c_{-2}c_{-1}$ (*Nb, Na*), c_1c_2 (*Ng, COMMA*), ... etc, associated with the category *VE* of target word '研究 research' from the following sentence:

周 Tsou (Nb) 先生 Mr (Na) 研究 research (VE) 之餘 after (Ng) ，(COMMA)

" After Mr. Tsou has done his research,"

By investigating all training data, different rule patterns will be generated, and their association probabilities $P(c_0' \mid w_0, \textit{feature vector})$ are also derived. For instance, If we take those word sequences

Word	Meaning	Characteristics
了	an expletive in the Chinese	high frequency
將	get, be about to	average distribution of candidate categories
研究	research	high inconsistence of context information
改變	change	simply two candidate categories
採訪	interview, gather material	low frequency
演出	perform	extreme low frequency

Table 2 Target words used in the experiments

listed in Table 1 as training data and $c_{-1}c_1$ as feature pattern, and set '.. research' as target word, we would train with a result containing a rule pattern = $c_{-1}c_1$ (*VH, PERIOD*) and derive the probabilities of $P(VE \mid \text{'..'}, (VH, PERIOD)) = 2/3$ and $P(NV \mid \text{'..'}, (VH, PERIOD)) = 1/3$. Suppose that the target word w_0 has ambiguous categories of $c_1, c_2, ..., c_n$, and the context patterns of $pattern_1, pattern_2, ..., pattern_m$, then the probability to assign tag c_i to the target word w_0 is defined as follows:

$$P(c_i) \cong \frac{\sum_{y=1}^{m} P(c_i \mid w, pattern_y)}{\sum_{x=1}^{n}\sum_{y=1}^{m} P(c_x \mid w, pattern_y)}$$

In other words, the probabilities of different patterns with the same candidate category are accumulated and normalized by the total probability distributed to all candidates as the probability of the candidate category. The algorithm will tag the category of the highest probability.

3 Experiment Results

The Sinica corpus is separated into two parts as our training data and testing data. The training data is randomly generated and utilizes 90% of the corpus, while the testing data is the remaining 10% part. Some ambiguous words' frequencies in the corpus are too low so that neither the context-rule algorithm nor the word-dependent Markov model is able to tag them well. Those words should be processed by other generic tagging algorithms. Therefore, we picked up words that its frequency is equal to or greater than 10 only as the target words in the experiments. The six ambivalence words with different frequencies, listed in Table 2, were picked as our example target words to see the performance of each tagging algorithm on words with different characteristics.

Some words like '.. interview' and '.. perform' have too low frequencies to have enough training data. To solve the problem of data sparseness, the Jeffreys-Perks law, or Expected Likehood Estimation (ELE), is introduced as the smoothing method for all evaluated tagging algorithms. To smooth for an unseen pattern $w_1, ..., w_n$, the probability $P(w_1, ..., w_n)$ is defined as $\frac{C(w_1, ..., w_n) + \lambda}{N + B\lambda}$, where $C(w_1, ..., w_n)$ is the amount that the pattern occurs in the training data, and N is the total amount of all training patterns, and B denotes the amount of all pattern types in training data and λ denotes the default occurrence count for an unseen pattern. The most widely used value for λ is 0.5, which is also applied in the experiments.

The Markov bi-gram model was evaluated to be compared with our context-rule model. Markov bi-gram model looks the category of the target word and categories before/after the target words. That is, given a word sequence $w_1, ...w_n$, it calculates the probability of each candidate category c_k for a target word w_k by $P(c_k \mid c_{k-1})\ P(c_{k+1} \mid c_k)\ P(w_k \mid c_k)$. In the experiments, we evaluate the probabilities of $P(c_k \mid c_{k-1})$ and $P(c_{k+1} \mid c_k)$ by two different approaches. One is to train all the sequences in the training data, and another one is to train only the sequences in which each w_k occurs.

Word	General Markov	Word-Depend. Markov	Context-Rule
了	96.95 %	97.92 %	98.87 %
將	93.47 %	93.17 %	95.52 %
研究	80.76 %	79.28 %	81.40 %
改變	87.60 %	89.92 %	93.02 %
採訪	68.06 %	63.89 %	77.78 %
演出	41.67 %	66.67 %	66.67 %
Avg. of 6 words	94.56 %	95.12 %	96.60 %
Avg. of all words	91.07 %	94.07 %	95.08 %

Table 3 Precision rates between evaluated tagging algorithms

The evaluated result is shown in table 3. The comparison of two approaches to evaluating $P(c_k \mid c_{k-1})$ and $P(c_{k+1} \mid c_k)$ in Markov model shows that using word-dependent context features is better than using all context features. The proposed context-rule model has higher precision rate than the Markov models.

4 Confidence Measure and Reduction on Manual Proofreading

The accuracy of a tagging result is usually estimated by the tagging precision of the algorithm. However the report precision of automatic tagging algorithm is about 95% to 96% (Chang et al., 1993; Lua, 1996; Liu et al., 1995). A better accuracy can be achieved if the tagging results are manually proofread. If we can pinpoint the errors, only 4~5% of the corpus has to be revised. Since it is not known where occurrences of errors are, conventionally the whole corpus has to be reexamined. It is most tedious and time consuming, since a practically useful tagged corpus is at least in the size of several million words. In order to reduce the manual editing and speed up the construction process of a large tagged corpus, a partial proofreading process has to be carried out. Only potential errors of tagging will be rechecked manually. The problem is how we find the potential errors of the tagging and what is a reliable tagging system, which can provide a confidence score for each step of tagging?

Since a probabilistic-based tagging method will assign a probability to each candidate pos-category, we assume that a candidate with higher probability might be more reliable. Therefore we adopt the following hypothesis. If the probability $P(c_1)$ of the top choice candidate c_1 is much higher than the probability $P(c_2)$ of the second choice candidate c_2, then the confidence value assigned for c_1 is also higher. Likewise if the probability $P(c_1)$ is closer to the probability $P(c_2)$, then the confidence value assigned for c_1 is also lower. A general confidence measure was defined as the value of $\frac{P(c_1)}{P(c_1)+P(c_2)}$, where $P(c_1)$ is the probability of the top choice category c_1 assigned by the tagging algorithm and $P(c_2)$ is the probability of the second choice category c_2. By using this definition of confidence measure, one can choose a confidence score, for example, 0.6, to filter those tagged words that have score lower than the pre-chosen confidence score, which are need manual proofreading. We like to prove the above hypothesis by empirical methods.

A tagging algorithm provided with a very reliable confidence score in some sense is a good cost-effective algorithm. A cost-effective algorithm may not be the algorithm with the highest precision. Therefore we defined below a new concept of reliability of a tagging system in term of cost-effective:

Reliability The estimated best accuracy can be achieved by the tagging model under the constraint that only a fixed amount of K% corpus with the lowest confidence value is manually proofread.

We carried out an experiment on the confidence measure on the context-rule tagging model. The target words are the ambivalence words of frequency greater than or equal to 10. Figure 1 shows the results. The confidence score is increased from 0.50, step in 0.01, to 1.00, to observe the curve between the amount of manual proofreading and the best accuracy with manual proofreading. When a certain confident score is chosen, some tagged words with confidence score lower than the chosen one will

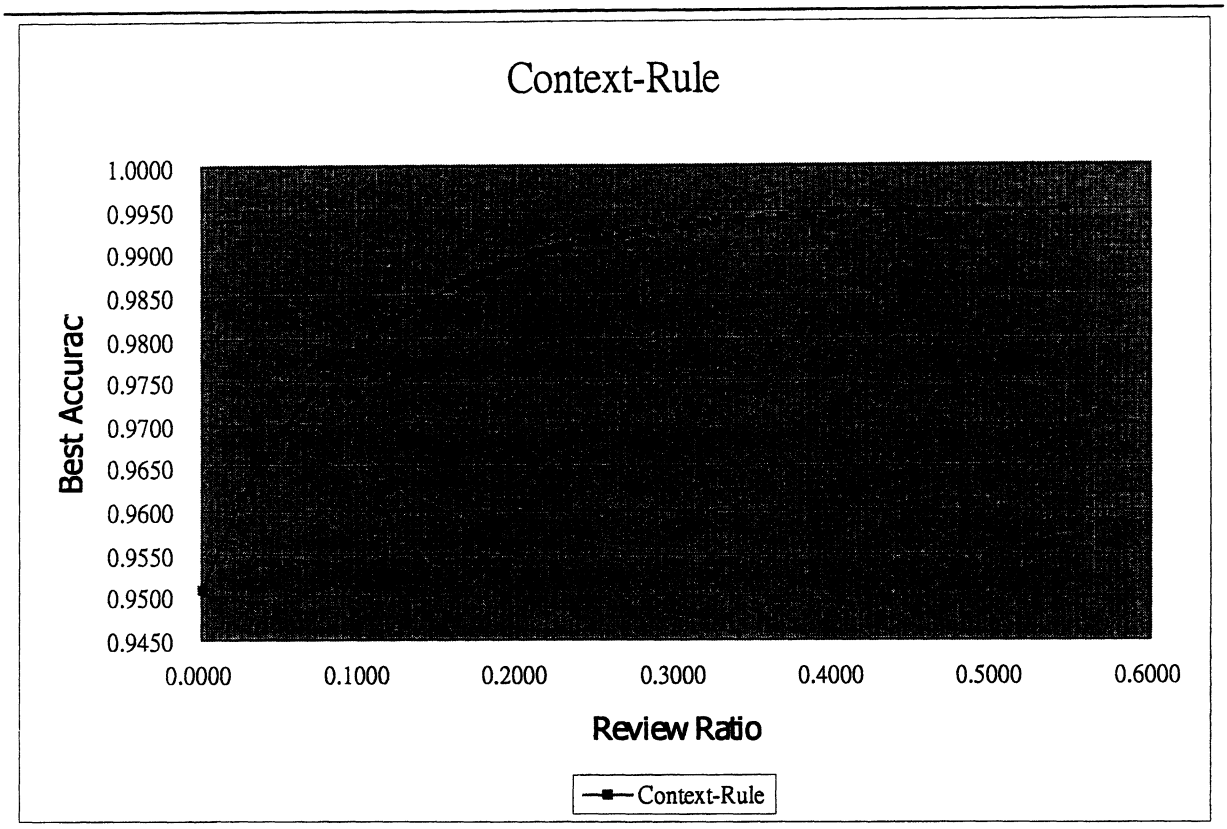

Figure 1 Tradeoffs between amount of manual proofreading and the best accuracy

show that they might have been tagged with wrong category. If those words are all manual proofread, the tagging accuracy can be increased more efficiently, and the final tagging accuracy can be easily estimated. For instance, if the confidence score is set 0.6, only 10.04% of tagged words have the confidence scores less than 0.6 that cover 57.92% error tagging. Therefore the estimated best tagging accuracy is 97.93%, if those tagged words with lower confidence score are all manual revised. The best accuracy is estimated by adding the amount of error reduction by manual proofreading to the original tagging accuracy, i.e. 95.08%+4.92%*57.92%=97.93% for confidence threshold of 0.6.

It is obviously that there is a trade-off between the accuracy and the amount of corpus to be manual proofread. The higher accuracy required, the larger corpus to be manual proofread. Thus, with a fixed resource of labour, one can determine the final accuracy of corpus after manual proofread is done, or he/she can estimate how many corpus should be manual proofread to achieve the required accuracy according to the curve.

5 Conclusion

The proposed context-rule model utilizes a broader scope of features to tag pos and achieve a better precision. The target word dependent Markov model also performs better than general Markov model. It clearly shown that to utilize more dependent features and more precise probability dependent statistics will perform better on the pos tagging. On the other hand the sparseness of training data reduces the accuracy of the tagging algorithm. Therefore use of more dependent features means more serious of data sparseness problem. However the context-rule model avoid the data sparseness problem by utilizing the rules with higher occurrence patterns only and use the general category patterns to cope with the low frequency target words. The context-rule tagging models focus on the ambivalence words only since top 300 ambivalence words contains 95% of tagging ambiguities according to Huang et al (2000). Therefore using confidence evaluation and context-rule models can drastically reduce amount of manual proofreading.

References

C. H. Chang & C. D. Chen, 1993, "HMM-based Part-of-Speech Tagging for Chinese Corpora," in Proceedings of the Workshop on Very Large Corpora, Columbus, Ohio, pp. 40-47.

C. J. Chen, M. H. Bai, & K. J. Chen, 1997, "Category Guessing for Chinese Unknown Words," in Proceedings of NLPRS97, Phuket, Thailand, pp. 35-40.

Christopher D. Manning & Hinrich Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999, pp. 202-204.

K. J. Chen, C. R. Huang, L. P. Chang, & H. L. Hsu, 1996, "Sinica Corpus: Design Methodology for Balanced Corpora," in Proceedings of PACLIC II, Seoul, Korea, pp. 167-176.

K. T. Lua, 1996, "Part of Speech Tagging of Chinese Sentences Using Genetic Algorithm," in Proceedings of ICCC96, National University of Singapore, pp. 45-49.

P. Kveton & K. Oliva, 2002, "(Semi-) Automatic Detection of Errors in PoS-Tagged Corpora," in Proceedings of Coling 2002, Taipei, Tai-wan, pp. 509-515.

S. H. Liu, K. J. Chen, L. P. Chang, & Y. H. Chin, 1995, "Automatic Part-of-Speech Tagging for Chinese Corpora," on Computer Proceeding of Oriental Languages, Hawaii, Vol. 9, pp.31-48.