Computational Linguistics and Chinese Language Processing
Vol. 9, No. 1 , February 2004, pp. 83-96

Reliable and Cost-Effective Pos-Tagging

Yu-Fang Tsai*, and Keh-Jiann Chen*

Abstract

In order to achieve fast, high quality Part-of-speech (pos) tagging, algorithms should achieve high accuracy and require less manually proofreading. This study aimed to achieve these goals by defining a new criterion of tagging reliability, the estimated final accuracy of the tagging under a fixed amount of proofreading, to be used to judge how cost-effective a tagging algorithm is. In this paper, we also propose a new tagging algorithm, called the context-rule model, to achieve cost-effective tagging. The context rule model utilizes broad context information to improve tagging accuracy. In experiments, we compared the tagging accuracy and reliability of the context-rule model, Markov bi-gram model and word-dependent Markov bi-gram model. The result showed that the context-rule model outperformed both Markov models. Comparing the models based on tagging accuracy, the context-rule model reduced the number of errors 20% more than the other two Markov models did. For the best cost-effective tagging algorithm to achieve 99% tagging accuracy, it was estimated that, on average, 20% of the samples of ambiguous words needed to be rechecked. We also compared tradeoff between the amount of proofreading needed and final accuracy for the different algorithms. It turns out that an algorithm with the highest accuracy may not always be the most reliable algorithm.

Keywords: part-of-speech tagging, corpus, reliability, ambiguous resolution

1. Introduction

Part-of-speech (pos) tagging for a large corpus is a labor intensive and time-consuming task. Most tagging algorithms try to achieve high accuracy, but 100% accuracy is an impossible goal. Even after tremendous amounts of time and labor are spent on the post-process of proofreading, many errors still exist in publicly available tagged corpora. Therefore, in order to achieve fast, high quality pos tagging, tagging algorithms should not only achieve high accuracy but also require less manually proofreading. In this paper, we propose a context-rule

* Institute of Information Science, Academia Sinica
128 Academia Rd. Sec.2, Nankang, Taipei, Taiwan E-mail: {eddie,kchen}@iis.sinica.edu.tw

model to achieve both goals.

The first goal is to improve tagging accuracy. According to our observation, the pos tagging of a word depends on its context but not simply on its context category. Therefore, the proposed context-rule model utilizes a broad scope of context information to perform pos tagging of a word. Rich context information helps to improve the model coverage rate and tagging accuracy. The context-rule model will be described in more detail later in this paper. Our second goal is to reduce the manual editing effort. A new concept of reliable tagging is proposed. The idea is as follows. An evaluation score is assigned to each tagging decision as an indicator of tagging confidence. If a high confidence value is achieved, it indicates that the tagging decision is very likely correct. On the other hand, a low confidence value means that the tagging decision requires manual checking. If a tagging algorithm can achieve a high degree of reliability in evaluation, this means that most of the high confidence tagging results need not manually rechecked. As a result, the time and manual efforts required in the tagging process can be drastically reduced. The reliability of a tagging algorithm is defined as follows:

Reliability = The estimated final accuracy achieved by the tagging model under the constraint that only a fixed number of target words with the lowest confidence values are manually proofread.

The notion of tagging reliability is slightly different from the notion of tagging accuracy since high accurate algorithm may require more manual proofreading than a reliable algorithm that achieves lower accuracy.

The rest of this paper is organized as follows. In section 2, the relation between reliability and accuracy is discussed. In section 3, three different tagging algorithms, the Markov pos bi-gram model, word-dependent Markov bi-gram model, and context-rule model, are discussed. In section 4, the three algorithms are compared based on tagging accuracy. In addition, confidence measures of tagging results are defined, and the most cost-effective algorithm is determined. Conclusions are drawn on section 5.

2. Reliability vs. Accuracy

The reported accuracy of automatic tagging algorithms ranges from about 95% to 96% [Chang *et al.*, 1993; Lua, 1996; Liu *et al.*, 1995]. If we can pinpoint errors, then only 4~5% of the target corpus has to be revised to achieve 100% accuracy. However, since the errors are not identified, conventionally, the whole corpus has to be re-examined. This is most tedious and time consuming since a practically useful tagged corpus is at least several million words in size. In order to reduce the amount manual editing required and speed up the process of constructing a large tagged corpus, only potential tagging errors should be rechecked manually [Kveton *et al.*, 2002; Nakagawa *et al.*, 2002]. The problem is how to find the potential errors.

Suppose that a probabilistic-based tagging method assigns a probability to each pos of a target word by investigating the context of this target word *w*. The hypothesis is that if the probability $P(c_1 \mid w, context)$ of the top choice candidate c_1 is much higher than the probability $P(c_2 \mid w, context)$ of the second choice candidate c_2, then the confidence value assigned to c_1 will also be higher. (Hereafter, for the purpose of simplification, we will use $P(c)$ to stand for $P(c \mid w, context)$, if without confusing.) Likewise, if the probability $P(c_1)$ is close to the probability $P(c_2)$, then the confidence value assigned to c_1 will also be lower. We aim to prove the above hypothesis by using empirical methods. For each different tagging method, we define its confidence measure according to the above hypothesis and examine whether tagging errors are likely to occur for words with low tagging confidence. If the hypothesis is true, we can proofread among the auto-tagged results only those words with low confidence values. Furthermore, the final accuracy of the tagging process after partial proofreading is done can also be estimated based on the accuracy of the tagging algorithm and the number of errors contained in the proofread data. For instance, suppose that a system has a tagging accuracy of 94%, and that K% of the target words with the lowest confidence scores covers 80% of the errors. After those K% of tagged words are proofread, 80% of the errors are fixed. Therefore, the reliability score of this tagging system of K% proofread words will be 1 - (error rate) * (reduced error rate) = 1 - ((1 - accuracy rate) * 20%) = 1 - ((1 - 94%) * 20%) = 98.8%. On the other hand, suppose that another tagging system has a higher tagging accuracy of 96%, but that its confidence measure is not very high, such that K% of the words with the lowest confidence scores contains only 50% of the errors. Then the reliability of this system is 1 - ((1 - 96%) * 50%) = 98%, which is lower than that of the first system. That is to say, after expending the same amount of effort on manual proofreading, the first system achieves better results even though it has lower tagging accuracy. In other words, a reliable system is more cost-effective.

3. Tagging Algorithms and Confidence Measures

In this paper, we will evaluate three different tagging algorithms based on the same training and testing data, compare them based on tgging accuracy, and determine the most reliable tagging algorithm among them. The three tagging algorithms are the Markov bi-gram model, word-dependent Markov model, and context-rule model. The training data and testing data were extracted from the Sinica corpus, a 5 million word balanced Chinese corpus with pos tagging [Chen *et al.*, 1996]. The confidence measure was defined for each algorithm, and the final accuracy was estimated with the constraint that only a fixed amount of testing data needed to be proofread.

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

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)
內(Ncd)	重點(Na)	**研究(Nv)**	需求(Na)	。(PERIODCATEGORY)
仍(D)	限於(VJ)	**研究(Nv)**	階段(Na)	。(PERIODCATEGORY)
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)	値得(VH)	**研究(VE)**	。(PERIODCATEGORY)	
合宜性(Na)	値得(VH)	**研究(VE)**	。(PERIODCATEGORY)	
更(D)	値得(VH)	**研究(Nv)**	。(PERIODCATEGORY)	

It is easier to proofread and obtain consistent tagging results if proofreading is done by checking each ambiguous word in its keyword-in-context file. For instance, in Table 1, the keyword-in-context file of the word '研究' (research), which has pos of verb type *VE* and noun type *Nv*, is sorted according to its left/right context. Proofreaders can take the other examples as references to determine whether tagging results are correct. If all of the occurrences of ambiguous words had to be rechecked, this would require too much work. Therefore, only words with low confidence scores will be rechecked.

A general confidence measure can be defined as $\frac{P(c_1)}{P(c_1)+P(c_2)}$, where $P(c_1)$ is the the probability of the top choice pos c_1 assigned by the tagging algorithm and $P(c_2)$ is the probability of the second choice pos c_2 [1]. The common terms used in the following tagging algorithms discussed below are defined as follows:

w_k the k-th word in a sequence;

c_k the pos associated with the k-th word w_k;

$w_1c_1,\ldots,w_nc_n$ a word sequence containing n words with their associated categories.

3.1 Markov Bi-gram Model

The most widely used tagging models are the part-of-speech n-gram models, in particular, the

[1] The log-likelihood ratio of log(P(c_1)/P(c_2)) is an alternative confidence measure. However, some tagging algorithms, such as context-rule model, may not necessary produce a real probability estimation for each pos. Scaling control for the log-likelihood ratio will be hard for those algorithms to achieve. In addition, the range of our confidence score is 0.5 ~ 1.0 and it is thus easier to evaluate different tagging algorithms. Therefore, the above confidence value is adopted.

bi-gram and tri-gram models. A bi-gram model looks at pairs of categories (or words) and uses the conditional probability of $P(c_k \mid c_{k-1})$. The Markov assumption is that the probability of a pos occurring depends only on the pos before it.

Given a word sequence $w_1,...w_n$, the Markov bi-gram model searches for the pos sequence $c_1,...c_n$ such that argmax $\Pi P(w_k \mid c_k) * P(c_k \mid c_{k-1})$ is achieved. In our experiment, since we were only focusing on the resolution of ambiguous words, a twisted Markov bi-gram model was applied. For each ambiguous target word, its pos with the highest model probability was tagged. The probability of each candidate pos c_k for a target word w_k was estimated as $P(c_k \mid c_{k-1}) * P(c_{k+1} \mid c_k) * P(w_k \mid c_k)$. We call this model the general Markov bi-gram model.

3.2 Word-Dependent Markov Bi-gram Model

The difference between the general Markov bi-gram model and the word-dependent Markov bi-gram model lies in the way in which the statistical data for $P(c_k \mid c_{k-1})$ and $P(c_{k+1} \mid c_k)$ is estimated. There are two approaches to estimating the probability. One is to count all the occurrences in the training data, and the other is to count only the occurrences in which each w_k occurs. In other words, the algorithm tags the pos c_k for w_k, such that c_k maximizes the probability of $P(c_k \mid w_k, c_{k-1}) * P(c_{k+1} \mid w_k, c_k) * P(w_k \mid c_k)$ instead of maximizing the probability of $P(c_k \mid c_{k-1}) * P(c_{k+1} \mid c_k) * P(w_k \mid c_k)$. We call this model the word-dependent Markov bi-gram model.

3.3 Context-Rule Model

The dependency features utilized to determine the best pos-tag in Markov models are the categories of context words. In fact, in some cases, the best pos-tags might be determined by using other context features, such as context words [Brill, 1992]. In the context-rule model, broad context information is utilized to determine the best pos-tag. We extend the scope of the dependency context of a target word to its 2 by 2 context windows. Therefore, the context features of a word 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 a unique pos-tag or many ambiguous pos-tags. The association probability of a possible pos c'_0 is $P(c'_0 \mid w_0$, *feature vector*). If for some (w_0, c'_0), the value of $P(c'_0 \mid w_0$, *feature vector*) is not 1, then this means that the c_0 of w_0 cannot be uniquely determined by its context vector. Some additional features have to be incorporated to resolve the ambiguity. If the full scope of the context feature vector is used, data sparseness problem will seriously degrade the system performance. Therefore, partial feature vectors are used instead of full feature vectors. The partial feature vectors applied in our context-rule model 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.

In the training stage, for each feature vector type, many rule instances are generated, and their probabilities associated with the pos of the target word are calculated. For instance, with the feature vector types w_{-1}, w_1, $c_{-2}c_{-1}$, c_1c_2,…, we can extract the 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 pos *VE* of the target word from the following sentence while the target word is '研究 research':

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

"After Mr. Tsou has done his research,"

Through the investigation of all training data, various different rule patterns (associated with a candidate pos of a target word) are generated and their association probabilities of *P*($c'_0 \mid w_0$, *feature vector*) derived. For instance, if we take those word sequences listed in 0 as training data and take $c_{-1}c_1$ as a feature pattern, and if we let '研究' be the target word, then the rule pattern $c_{-1}c_1$(*VH*, *PERIOD*) will be extracted, and we will derive the probabilities *P*(*VE* | '研究', (*VH*, *PERIOD*)) = 2/3 and *P*(*NV* | '研究', (*VH*, *PERIOD*)) = 1/3. The rule patterns and their association probability are used to determine the probability of each candidate pos of a target word in a testing sentence. Suppose that the target word w_0 has ambiguous categories $c_1, c_2, ..., c_n$, and context patterns *pattern*$_1$, *pattern*$_2$, …, *pattern*$_{\mathrm{m}}$; then, the probability of assigning 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 pos are accumulated and normalized by means of the total probability distributed to all the candidates as the probability of the candidate pos. The algorithm tags the pos of the highest probability.

4. Experiments and Results

For our experiments, the Sinica corpus was divided into two parts. The training data contained 90% of the corpus, while the testing data contained the remaining 10%. Only the target words with ambiguous pos were evaluated. We evaluated only on the ambiguous words with frequencies higher than or equal to 10 for sufficiency of the training data and testing data. Furthermore, the total token count of the words with frequencies less than 10 occupied only 0.4335% of all the ambiguous word tokens. Since those words had much less effect on the overall performance, we just ignored them to simplify the designs of the evaluated tagging systems in the experiments. Another important reason was that for those words with low frequencies, all their tagging results had to be rechecked anyway, since our experiments showed that low tagging accuracies were inevitable due to the lack of training data. We also examined the effects on the tagging accuracy and reliability on the words with variations on pos ambiguities and the amount of training data. Six ambiguous words with different frequencies, listed in Table 2, were selected as our target words for detail examinations.

Table 2. Target words used in the experiments tagging accuracy.

Word	Frequency	Ambiguity (Pos-Count)				
了	47607	Di-36063	T-11504	VJ-25	VC-11	
將	13188	D-7599	P-5547	Na-27	Di-8	VC-5
研究	4734	Nv-3695	VE-1032	VC-6	VA-1	
改變	1298	VC-953	Na-345			
演出	723	VC-392	Na-331			
採訪	121	VC-70	Nv-45	Na-6		

Table 3. Accuracy rates of the evaluated tagging algorithms.

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 %
Average of 6 words	94.56 %	95.12 %	96.60 %
Average of all ambiguous words	91.07 %	94.07 %	95.08 %

The frequencies of some words were too low to provide enough training data, such as the words '採訪 interview' and '演出 perform' listed in 0. To solve the problem of data sparseness, the Jeffreys-Perks law, or Expected Likehood Estimation (ELE) [Manning *et al.*, 1999], was used as a smoothing method for all the tagging algorithms evaluated in the experiments. The probability $P(w_1,..., w_n)$ was defined as $\frac{C(w_1,..., w_n)}{N}$, where $C(w_1,..., w_n)$ is the number of times that pattern $w_1,..., w_n$ occurs in the training data, and N is the total number of training patterns. To smooth for an unseen event, the probability of $P(w_1,..., w_n)$ was redefined as $\frac{C(w_1,..., w_n)+\lambda}{N+B\lambda}$, where B denotes the number of all pattern types in the training data and λ denotes the default occurrence count for an unseen event. That is to say, we took a value λ for an unseen event as its occurrence count. If the value of λ was 0, this means that there was no smoothing process for the unseen event. The most widely used value for λ is 0.5, which was also applied in our experiments.

4.1 Tagging Accuracy

In the experiments, we compared the tagging accuracy of the three tagging algorithms as described in section 3. The experiment results are shown in Table 3. It is obvious that the word-dependent Markov bi-gram model outperformed the general Markov bi-gram model. It reduced almost 30% the number of errors compared to the general Markov bi-gram model. As expected, the context-rule model performed the best for each selected word and the overall tagging accuracy. The tagging accuracy results for selected words show inconsistency. This is exemplified by the lower accuracy for the word '研究 research'. It is believed that the flexible usage of '研究 research' degraded the performances of the tagging algorithms. The lack of training data also hurt the performance of the tagging algorithms. The words with fewer training data, such as '採訪 interview' and '演出 perform', were also associated with poor tagging accuracy. Therefore, words with low frequencies should be handled using some general tagging algorithms to improve the overall performance of a tagging system. Furthermore, in future, word-dependent reliability criteria need to be studied.

4.2 Tagging Reliability

In the experiments on reliability, the confidence measure of the ratio of the probability gap between the top choice candidate and the second choice candidate $\frac{P(c_1)}{P(c_1)+P(c_2)}$ was adopted for all three models. The tagging results with confidence scores lower than a pre-defined threshold were re-checked. Some tagging results were assigned the default pos (in general, the one with the highest frequency of the word) since there were no training patterns applicable to the tagging process. Those tagging results that were not covered by the training patterns also needed to be re-checked. With the increased pre-defined threshold, the amount of partial corpus that needed to be re-checked could be estimated automatically since the Sinica corpus provides the correct pos-tag for each target word. Furthermore, the final accuracy could be estimated if the corresponding amount of partial corpus was proofread.

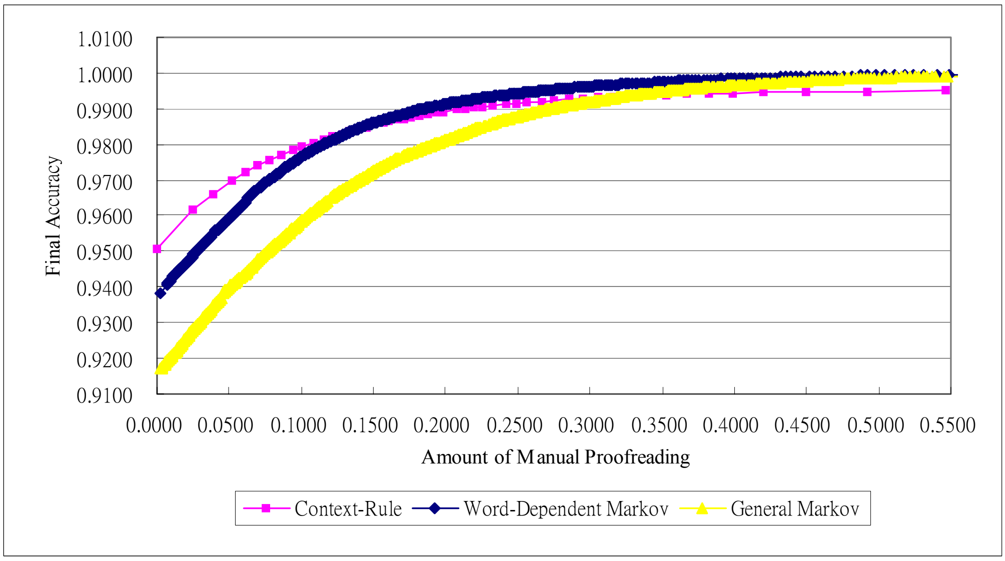

Figure 1. Tradeoff between the amount of manual proofreading and the final accuracy.

Figure 1 shows the results for the tradeoff between the amount of proofreading and the estimated final accuracy for the three algorithms. The x-coordinate indicates the portion of the partial corpus that needed to be manually proofread under a pre-defined threshold. The y-coordinate indicates the final accuracy after the corresponding portion of the corpus was proofread. Without any manual proofreading, the accuracy of the context-rule algorithm was about 1.4% higher than that of the word-dependent Markov bi-gram model. As the percentage of manual proofreading increased, the accuracy of each algorithm also increased. It is obvious that the accuracy of the context-rule model increased more slowly than did that of the two Markov models, as the amount of manual proofreading increased.

The final accuracy results of the context-rule model and the two Markov models coincided at approximately 98.5% and 99.4%, with around 13% and 35% manual proofreading. After that, both Markov models achieved higher final accuracy than the context-rule model did when the amount of manual proofreading increased more. The results indicate that if the required tagging accuracy is over 98.5%, then the two Markov models will be better choices since in our experiments, they achieved higher final accuracy than the context-rule model did. It can also be concluded that an algorithm with higher accuracy may not always be an accurate algorithm.

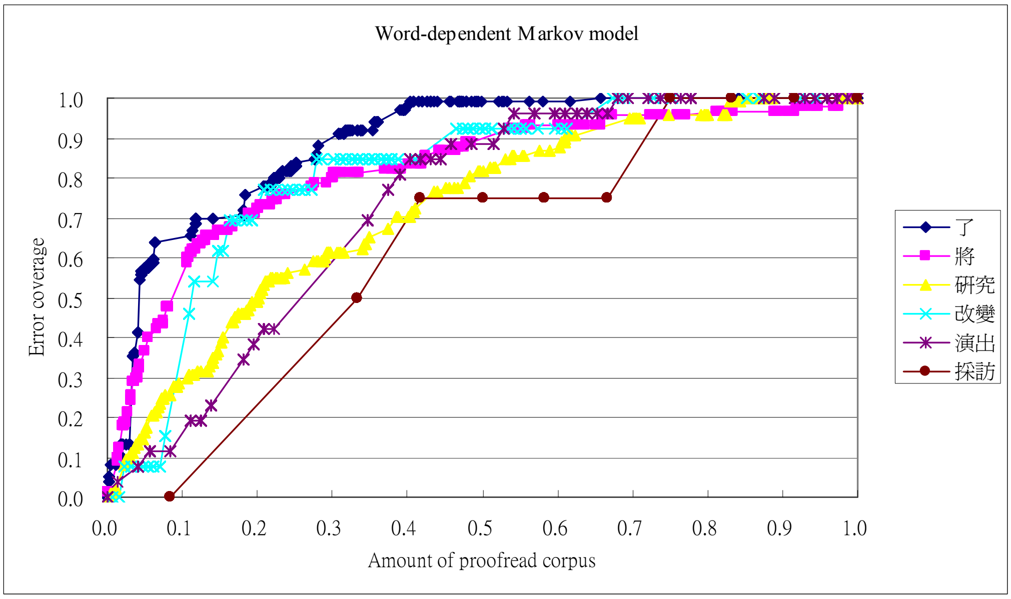

Figure 2. Error coverage of word-dependent Markov model after amount of corpus is proofread.

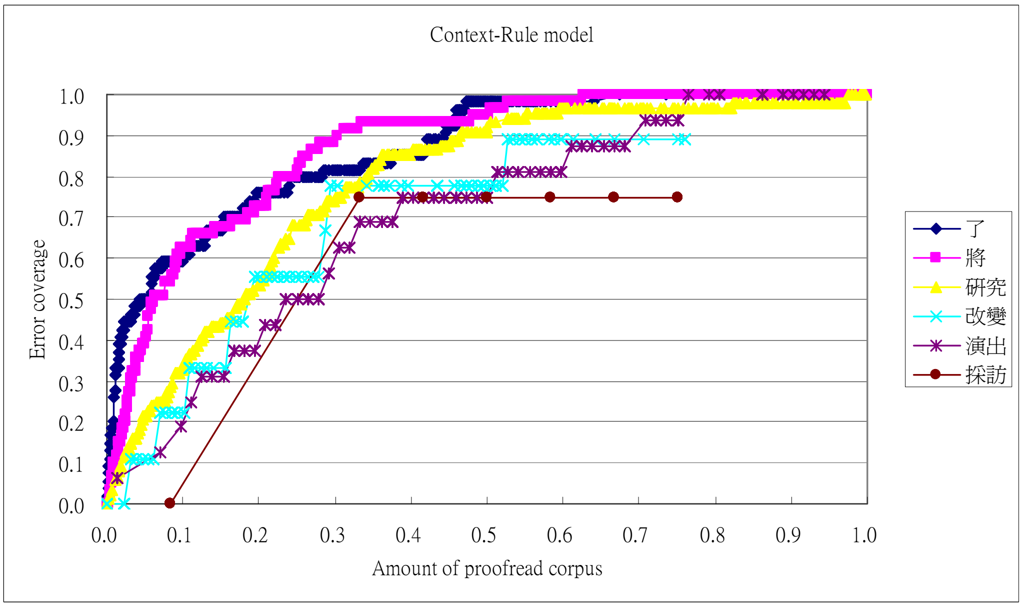

Figure 3. Error coverage of context-rule model after amount of corpus is proofread.

Figure 2 and Figure 3 show the error coverage of the six ambiguous target words after different portions of corpus are proofread respectively. It shows that not only tagging accuracy but also reliability were degraded due to the lack of sufficient training data. Tagging algorithms achieve better error coverage for target words with more training data.

4.3 The Tradeoff between the Amount of Manual Proofreading and the Final accuracy

There is a tradeoff between amount of manual proofreading and the final accuracy. If the goal of tagging is to achieve 99% accuracy, then an estimated threshold value of the confidence score needed to achieve the target accuracy rate will be given, and a tagged word with a confidence score less than this designated threshold value will be checked. On the other hand, if the requirement is to finish the tagging process in a a limited amount of time and with limited amount of manual labor, then in order to achieve the desired final accuracy, we will first need to estimate the portion of the corpus which will have to be proofread, and then determine the threshold value of the confidence score. Figure 4 shows the error coverage of each different portions of corpus with the lowest confidence score. By proofreading the initial 10% low confidence tagging data we achieve the most improvement in accuracy. As the amount of proofread corpus increased, the level of accuracy decreased rapidly. The experimental results of tagging reliability can help us decide which is the most cost-effective tagging algorithm and how to proofread tagging results under constraints on the available human resources and time.

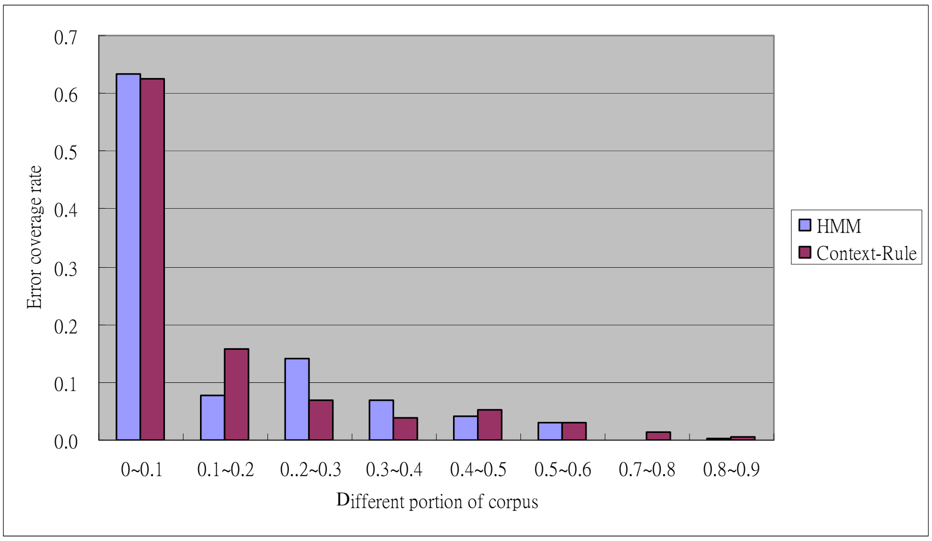

Figure 4. Error coverage rate of different portion of corpus to be proofread.

5. Conclusion

In this paper, we have proposed a context-rule model for pos tagging. We have also proposed a new way of finding the most cost-effective tagging algorithm. Cost-effectiveness is defined based on a criterion of reliability. The reliability of the system is measured in terms of the confidence score for ambiguity resolution of each tagging. The basic observation of confidence tagging is as follows: the larger the gap between the candidate pos with the highest probability and other (the second, for example) candidate pos with lower probability, the more confidence can be placed in the tagging result. It is believed that the ability to resolve pos ambiguity plays a more important part than the confidence measurement in the tagging system, since a larger gap between the first candidate pos and the second candidate pos can result in a high confidence score. Therefore, another reasonable measurement of the confidence score will work as well as the one used in our experiments if the tagging algorithms have good ability to resolve pos ambiguity.

For the best cost-effective tagging algorithm, on average, 20% of the samples of ambiguous words need to be rechecked to achieve 99% accuracy. In other words, the manual labor of proofreading is reduced by more than 80%. Our study on tagging reliability, in fact, provides a way to determine the optimal tagging strategy under different constraints. The constraints might be to achieve the best tagging accuracy under time and labor constraints or to achieve a certain accuracy with the least effort possible expended on proofreading. For instance, if the goal of tagging is to achieve 99% accuracy, then a threshold value of the confidence score needed to achieve the target accuracy will be estimated, and a tagged word with a confidence score less than this designated threshold value will be checked. On the other hand, if the constraint is to finish the tagging process under time and manual labor constraints, then in order to achieve the desired final accuracy, we will first estimate the portion of the corpus that will have to be proofread, and then determine the threshold value of the confidence score.

In future, we will extend the coverage of confidence checking for all words, including words with single pos, to detect flexible word usages. The confidence measure for words with single pos can be obtained by comparing the tagging probability of the pos of the words with the probabilities of the other categories. Furthermore, since tagging accuracy and reliability are degrading due to the intrinsic complexity of word usage and the less amount of training data, we will study word-dependent reliability to overcome the degrading problems. There are many possible confidence measures. For instance $\log(p(c_1)/p(c_2))$ is a reasonable alternative. We will study different alternatives in the future to obtain a more reliable confidence measure.

Acknowledgement:

The work was partially supported under NSC grant 92-2213-E-001-016.

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. 43-45, pp. 202-204.

E. Brill, "A Simple Rule-Based Part-of-Speech Taggers," in Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing 1992, pp. 152–155.

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, Taiwan, 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.

T. Nakagawa & Y. Matsumoto, 2002, "Detecting Errors in Corpora Using Support Vector Machines," in Proceedings of Coling 2002, Taipei, Taiwan, pp.709-715.