Reliable and Cost-Effective PoS-Tagging

Yu-Fang Tsai **Keh-Jiann Chen**
Institute of Information Science, Academia Sinica
Nankang, Taipei, Taiwan 115
eddie,kchen@iis.sinica.edu.tw

Abstract

In order to achieve fast and high quality Part-of-speech (PoS) tagging, algorithms should be high accuracy and require less manually proofreading. To evaluate a tagging system, we proposed a new criterion of reliability, which is a kind of cost-effective criterion, instead of the conventional criterion of accuracy. The most cost-effective tagging algorithm is judged according to amount of manual editing and achieved final accuracy. The reliability of a tagging algorithm is defined to be the estimated best accuracy of the tagging under a fixed amount of proofreading.

We compared the tagging accuracies and reliabilities among different tagging algorithms, such as Markov bi-gram model, Bayesian classifier, and context-rule classifier. According to our experiments, for the best cost-effective tagging algorithm, in average, 20% of samples of ambivalence words need to be rechecked to achieve an estimated final accuracy of 99%. The tradeoffs between amount of proofreading and final accuracy for different

algorithms are also compared. It concludes that an algorithm with highest accuracy may not always be the most reliable algorithm.

1 Introduction

Part-of-speech tagging for a large corpus is a labor intensive and time-consuming task. Most of time and labors were spent on proofreading and never achieved 100% accuracy, as exemplified by many public available corpora. Since manual proofreading is inevitable, how do we derive the most cost-effective tagging algorithm? To reduce efforts of manual editing, a new concept of reliable tagging was proposed. The idea is as follows. An evaluation score, as an indicator of tagging confidence, is made for each tagging decision. If a high confidence value is achieved, it indicates that this tagging decision is very likely correct. On the other hand, a low confidence value means the tagging result might require manual checking. If a tagging algorithm can provide a very reliable confidence evaluation, it means that most of high confidence tagging results need not manually checked. As a result, time and manual efforts for tagging processes can be reduced drastically. 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 amount target words with the lowest confidence value is manually proofread.

It is slightly different from the notion of tagging accuracy. It is possible that a higher accuracy algorithm might require more manual proofreading than a reliable algorithm with lower accuracy.

The tagging accuracies were compared among different tagging algorithms, such as Markov PoS bi-gram model, Bayesian classifier, and context-rule classifier. In addition, confidence measures of the tagging will be defined. In this paper, the above three algorithms are designed and the most cost-effective algorithm is also determined.

2 Reliability vs. Accuracy

The reported accuracies of automatic tagging algorithms are about 95% to 96% (Chang et al., 1993; Lua, 1996; Liu et al., 1995). If we can pinpoint the errors, only 4~5% of the target corpus has to be revised to achieve 100% accuracy. However, since the occurrences of errors are unknown, 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, only potential errors of tagging will be rechecked manually (Kveton et al., 2002; Nakagawa et al., 2002). The problem is how we find the potential errors. Suppose that a probabilistic-based tagging method will assign 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 for c_1 is also higher. (Hereafter, for simplification, if without confusing, we will use $P(c)$ to stand for $P(c \mid w, context)$.) 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. We try to prove the

above hypothesis by empirical methods. For each different tagging method, we define its confidence measure according to the above hypothesis and to see whether or not tagging errors are generally occurred at the words with low tagging confidence. If the hypothesis is true, we can proofread the auto-tagged results only on words with low confidence values. Furthermore, the final accuracy of the tagging after partial proofreading can also be estimated by the accuracy of the tagging algorithm and the amount of errors contained in the proofread data. For instance, a system has a tagging accuracy of 94% and supposes that K% of the target words with the lowest confidence scores covers 80% of errors. After proofreading those K% of words in the tagged words, those 80% errors are fixed. Therefore the reliability score of this tagging system of K% proofread will be 1 - (error rate) * (reduced error rate) = 1 - ((1 - accuracy rate) * 20%) = 1 - ((1 - 94%) * 20%) = 0.988. On the other hand, another tagging system has a higher tagging accuracy of 96%, but its confidence measure is not very reliable, such that the K% of the words with the lowest confidence scores contains only 50% of errors. Then the reliability of this system is 1 - ((1 - 96%) * 50%) = 0.980, which is lower than the first system. That is to say after spending the same amount of effort of manual proofreading, the first system achieves a better results even it has lower tagging accuracy. In other word, a reliable system is more cost-effective.

3 Tagging Algorithms and Confidence Measures

In this study, we are going to test three different tagging algorithms based on same training data and testing data, and to find out the most reliable tagging algorithm. The three tagging algorithms are

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)	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

Markov bi-gram model, Bayesian classifier, and context-rule classifier. 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 PoS 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 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 correct. If all of the occurrences of ambivalence word have to be rechecked, it is still too much of the work. Therefore only words with low confidence scores will be rechecked.

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 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 were also defined as follows:

w_k	The k-th word in a sequence
c_k	The PoS associated with k-th word w_k
$w_1c_1,...,w_nc_n$	A word sequence containing n words with their associated categories respectively

3.1 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 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 are only focusing on the resolution of ambivalence words only, a twisted Markov bi-gram model was

[1] Log-likelihood ratio of log P(c1)/P(c2) is another alternation of confidence measure. However, for some tagging algorithms, they may not necessary produce real probability estimation for each PoS, such as context-rule model. The scaling control for log-likelihood ratio will be hard for those algorithms. In addition, the range of our confidence score is between 0.5~1.0. Therefore, the above confidence value is adopted.

applied. For each ambivalence target word, its PoS with the highest model probability is tagged. The probability of each candidate PoS c_k for a target word w_k is estimated by $P(c_k \mid c_{k-1}) * P(c_{k+1} \mid c_k) * P(w_k \mid c_k)$. There are two approaches to estimate the statistical data for $P(c_k \mid c_{k-1})$ and $P(c_{k+1} \mid c_k)$. One is to count all the occurrences in the training data, and another one is to count only the occurrences in which each w_k occurs. According to the experiments, to estimate the statistic data using w_k dependent data is better than using all sequences. 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 $P(c_k \mid c_{k-1}) * P(c_{k+1} \mid c_k) * P(w_k \mid c_k)$.

3.2 Bayesian Classifier

The Bayesian classifier algorithm adopts the Bayes theorem (Manning et al., 1999) that swaps the order of dependence between events. That is, it calculates $P(c_{k-1} \mid c_k)$ instead of $P(c_k \mid c_{k-1})$. The probability of each candidate PoS c_k in Bayesian classifier is calculated by $P(c_{k-1} \mid w_k, c_k) * P(c_{k+1} \mid w_k, c_k) * P(c_k \mid w_k)$. The Bayesian classifier tags the PoS c_k for w_k, such that c_k maximizes the probability of $P(c_{k-1} \mid w_k, c_k) * P(c_{k+1} \mid w_k, c_k) * P(c_k \mid w_k)$.

3.3 Context-Rule Model

Dependency features utilized in determining the best PoS-tag in both Markov and Bayesian models are categories of context words. As a matter of fact, for some cases the best PoS-tags might be determined by other context features, such as context words (Brill, 1992). In the context-rule model,

broader scope of context information is utilized in determining the best PoS-tag. 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 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. Their association probability of a possible PoS 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 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 for a 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 and their probabilities associated with PoS of the target word are also calculated. For instance, with the feature vector types of w_{-1}, w_1, $c_{-2}c_{-1}$, c_1c_2,…, 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 PoS *VE* of 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,”

By investigating all training data, various different rule patterns (associated with a candidate PoS of a target word) will be generated and their association probabilities of $P(c_0' \mid w_0$, *feature vector*) are also derived. For instance, If we take those word sequences listed in Table 1 as training data and $c_{-1}c_1$ as feature pattern, and set ‘研究’ 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* | ‘研究’, (*VH*, *PERIOD*)) = 2/3 and *P*(*NV* | ‘研究’, (*VH*, *PERIOD*)) = 1/3. The rule patterns and their association probability will be utilized 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 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 PoS are accumulated and normalized by the total probability distributed to all candidates as the probability of the candidate PoS. The algorithm will tag the PoS of the highest probability.

4 Tradeoffs between Amount of Manual Proofreading and the Best Accuracy

There is a tradeoff between amount of manual proofreading and the best accuracy. If the goal of tagging is to achieve an accuracy of 99%, then an estimated threshold value of confidence score to

Word	Word Sense	Distribution 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	extremely low frequency

Table 2 Target words used in the experiments

achieve the target accuracy will be given and the tagged word with 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 the constraints of limited time and manual labors, in order to achieve the best accuracy, we will first estimate the amount of partial corpus which can be proofread under the constrained time and labors, and then determine the threshold value of the confidence.

The six ambivalence words with different frequencies, listed in Table 2, were picked as our target words in the experiments. We like to see the tagging accuracy and confidence measure effected by variation of ambivalence and the amount of training data among selected target words. The Sinica corpus is divided into two parts as our training data and testing data. The training data contains 90% of the corpus, while the testing data is the remaining 10%.

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

In our experiments, the confidence measure of the ratio of probability gap between top choice candidate and the second choice candidate $\frac{P(c_1)}{P(c_1)+P(c_2)}$ is adopted for all three different models. Figure 1 shows the result pictures of tradeoffs between amount of proofreading and the estimated best accuracies for the three different algorithms. Without any manual proofreading on result tags, the accuracy of context-rule algorithm is about 1.4% higher than the Bayesian classifier and Markov bi-gram model. As the percentage of manual proofreading increases, the accuracy of each algorithm increases, too. It is obvious to see that the accuracy of context-rule algorithm increases slower than those of other two algorithms while the amount of manual proofreading increases more. The values

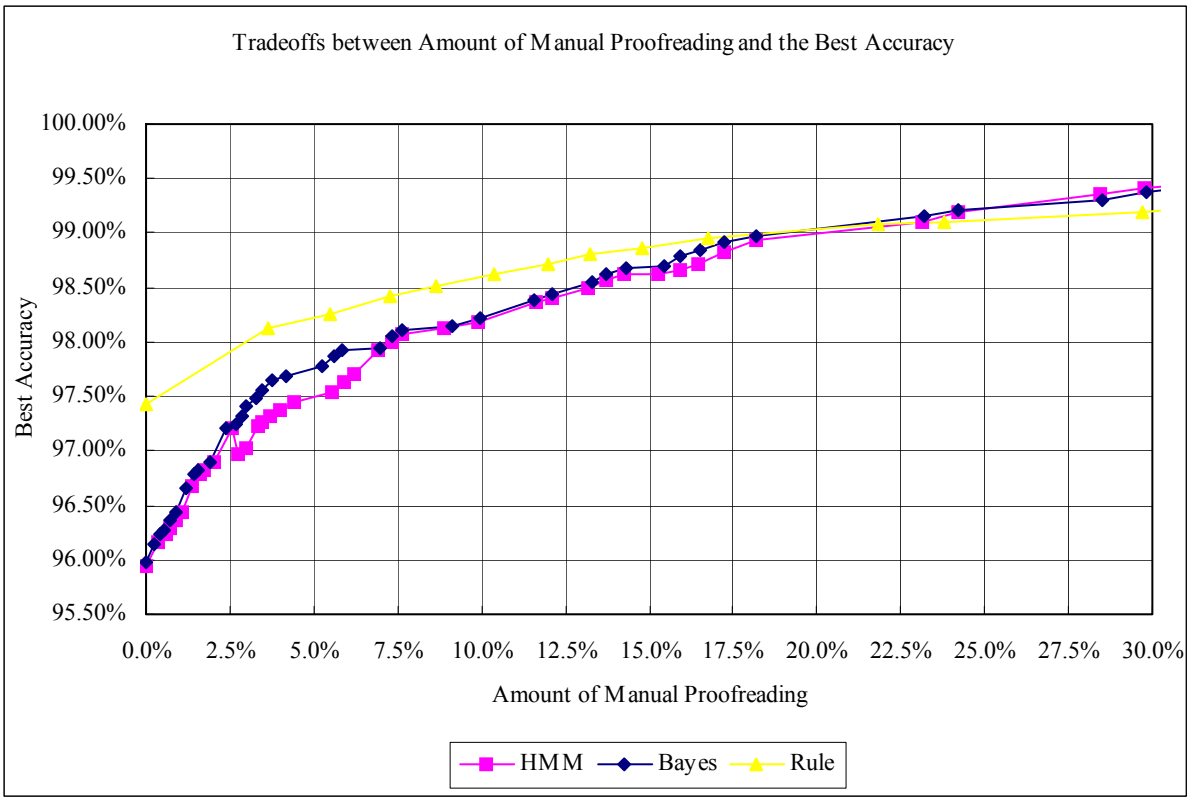

Figure 1 Tradeoffs between amount of manual proofreading and the best accuracy

of best accuracy of three algorithms will meet in a point of 99% approximately, with around 20% of required manual proofreading on result tags. After the meeting point, Bayesian classifier and Markov bi-gram model will have higher value of best accuracy than context-rule classifier when the amount of manual proofreading is over 20% of the tagged results.

The result picture shows that if the required tagging accuracy is over 99% and there are plenty of labors and time available for manual proofreading, the Bayesian classifier and Markov bi-gram model would be better choices, since they have higher best accuracies than the context-rule classifier.

5 Conclusion

In this paper, we proposed a new way of finding the most cost-effective tagging algorithm. The cost-effective is defined in term of a criterion of reliability. The reliability of the system is measured in term of confidence score of ambiguity resolution of each tagging. For the best cost-effective tagging algorithm, in average, 20% of samples of ambivalence words need to be rechecked to achieve an accuracy of 99%. In other word, the manual labor of proofreading is reduced more than 80%.

In future, we like to extend the coverage of confidence checking for all words, including words with single PoS, to detect flexible word uses. The confidence measure for words with single PoS can be made by comparing the tagging probability of this particular PoS with all other categories.

Acknowledgement: The authors would like to thank the anonymous reviews' valuable comments and suggestions. The work is partially supported by the grant of NSC 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, 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.

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