Computational Linguistics and Chinese Language Processing
Vol. 13, No. 2, June 2008, pp. 157-170

A Study on Consistency Checking Method of Part-Of-Speech Tagging for Chinese Corpora[1]

Hu Zhang[*] and Jiaheng Zheng[*]

Abstract

Ensuring consistency of Part-Of-Speech (POS) tagging plays an important role in the construction of high-quality Chinese corpora. After having analyzed the POS tagging of multi-category words in large-scale corpora, we propose a novel classification-based consistency checking method of POS tagging in this paper. Our method builds a vector model of the context of multi-category words along with using the *k*-NN algorithm to classify context vectors constructed from POS tagging sequences and to judge their consistency. These methods are evaluated on our 1.5M-word corpus. The experimental results indicate that the proposed method is feasible and effective.

Keywords: Multi-Category Words, Consistency Checking, Part of Speech Tagging, Chinese Corpus, Classification

1. Introduction

The construction of high-quality and large-scale corpora has always been a fundamental research area in the field of Chinese natural language processing. In recent years, rapid developments in the fields of machine translation (MT), information retrieval (IR), etc. are demanding more Chinese corpora of higher quality and larger scale. Ensuring the consistency of Part-of-Speech (POS) tagging plays an important role in the construction of high-quality Chinese corpora. In particular, we focus on consistency checking of the POS tagging of multi-tagged words, which consist of same Chinese characters and are nearly synonymous, yet have different grammatical functions. No matter how many different POS tags a multi-category word may be tagged with, it must be assigned the same POS tag when it appears in a similar context.

[1] This research was partially supported by the National Natural Science Foundation of China No. 60473139, 60775041 and the Natural Science Foundation of Shanxi Province No. 20051034.
[*] School of Computer & Information Technology, Shanxi University, Taiyuan 030006, China
Tel: +86-3517010566
E-mail: {zhanghu; jhzheng}@sxu.edu.cn

[Received January 16, 2007; Revised August 21, 2007; Accepted September 3, 2007]

The general process of tagging text in the Chinese corpora consists of two steps: (1) automatic POS tagging is used to generate preliminary POS tags; (2) the preliminary POS tags are checked manually by domain experts. Novel approaches and techniques have been proposed for automatic rule-based and statistics-based POS tagging, and the "state-of-the-art" approaches achieve a tagging precision of at least 95% [Zhang *et al.* 1998; Huang *et al.* 2004; Xue *et al.* 2002; Ng *et al.* 2004]. A vast proportion of the words appearing in Chinese corpora are multi-category words. We have studied the textual data from the 2M-word Chinese corpus published by Peking University, and statistics show that the number of multi-category words is 5,473, covering 11% of the word types, while the number of word tokens is as high as 438,624, the percentage of which is 47%. Relevant works indicate that the different POS tag sets have different Stat. Huang *et al.* [2004] provided some important facts based on the 5M-word Sinica Corpus, wherein categorically ambiguous words only take up only 4.298% of all lexical items; however, categorically ambiguous words compose over 54.5% of all tokens. When checking the POS tags, human experts may have disagreements or may make mistakes in some cases. After analyzing 10,000 sentences containing the multi-category words, which were extracted from the 2M-word Chinese corpus of Peking University, the number of incorrect tags for multi-category words was found to be 133, which accounts for around 1.33% of the total.

So far in the field of POS tagging, most of the works have focused on novel algorithms or techniques for POS tagging. There are only a limited number of studies that have focused on consistency checking of POS tagging. Xing *et al.* [1999] analyzed the inconsistency phenomena of word segmentation (WS) and POS tagging. Qu and Chen [2003] improved the corpus quality by obtaining POS tagging knowledge from processed corpora. Qian and Zheng [2003] introduced a rule-based consistency checking method that obtained POS tagging knowledge automatically from processed corpora by machine learning (ML) and rough set (RS) methods. For real corpora, Du and Zheng [2001] proposed a rule-based consistency checking method to identify the inconsistency phenomena of POS tagging. However, the algorithms and techniques for automatic consistency checking of POS tagging proposed in the prior researches still have some insufficiencies. For example, the assignment of POS tags of the inconsistent POS tagging that are not included in the instance set needs to be conducted manually.

In this paper, we propose a novel classification-based method to check the consistency of POS tagging. Compared to Zhang *et al.* [2004], the proposed method fully considers the mutual relationships of the POS in POS tagging sequence, and adopts transition probability and emission probability to describe the mutual dependencies and the *k*-NN algorithm to weight the similarity. We evaluated our proposed algorithm on our 1.5M-word corpus. In an open test, our method achieved a precision rate of 85.24% and a recall rate of 85.84%.

The rest of the paper is organized as follows. Section 2 introduces the context vector model of POS tagging sequences. Section 3 describes the proposed classification-based consistency checking algorithm. Section 4 discusses the experimental results. Finally, the concluding remarks are given in Section 5.

2. Describing the Context of Multi-Category Words

The basic idea of our approach is to use the contextual information of multi-category words to judge whether they are tagged consistently or not. In other words, if a multi-category word appears in two locations and the surrounding words in those two locations are tagged similarly, the multi-category word should be assigned with the same POS tag in those two locations as well. Hence, our approach is based on the context of multi-category words, and we model the context by looking at a window around a given multi-category word and the tagging sequence of this window. In the rest of this section, we describe the vector representation of the context of multi-category words and how to determine various parameters in our vector representations.

2.1 Vector Representation of the Context of Multi-Category Words

Our vector representation of context consists of three key components: the POS tags of each word in a context window (*POS attribute*), the importance of each word to the center multi-category word based on distance (*position attribute*), and the dependency of POS tags of the center multi-category word and its surrounding words (*Dependency Attribute*).

Given a multi-category word and its context window of size *l*, we represent the words in sequential order as $(w_1,w_2,\ldots,w_l)$ and the POS tags of each word as $(t,t_2,\ldots,t_l)$. We also refer to the latter vector as POS *tagging sequence*.

In practice, we choose a proper value of *l* so that the context window contains a sufficient number of words and the complexity of our algorithm remains relatively low. We will discuss this matter in detail later. In this study, we set the value of *l* to be 7, which means w_4 is our multi-category word of interest and the context window includes 3 preceding and following words, where w_{4-i} is the *i*th preceding word and w_{4+i} is the *i*th following word.

POS Attribute

The POS tagging sequence contains information of the POS of each preceding and following word in a POS tagging sequence as well as the position of each POS tag. The POS of surrounding words may have different effect on determining the POS of the multi-category word, which we refer to as *POS attribute* and represent it using a matrix as follows.

Definition 1. *Suppose we have a tag set of size* $m(c_1,c_2,\ldots,c_m)$*. Given a multi-category word with a context window of size* l $(w_1,w_2,\ldots,w_l)$ *and its POS tagging sequence* $(t,t_2,\ldots,t_l)$*, the POS*

attribute matrix Y is an l by m matrix, where the rows indicate the POS tags of the preceding words, the multi-category word, and the following words in the context window, while the columns present tags in the tag set. $Y_{i,j}=1$ *iff the POS tag of* t_i *is* c_j.

For example, consider the POS attribute matrix of "比较" in the following sentence:

我们/r 就/d 能/v 比较/d 准确/a 地/u 预测/v 出/v 队员/n 的/u 成绩/n

As we let l=7, we look at the word "比较" and its 3 preceding and following words. Hence, the POS tagging sequence is (*r,d,v,d,a,u,v*). In our study, we used a standard tag set that consists of 25 tags. Suppose the tag set is (*n, v, a, d, u, p, r, m, q, c, w, I, f, s, t, b, z, e, o, l, j, h, k, g, y*), then the POS attribute matrix of "比较" in this example is:

$$\boldsymbol{Y}=\begin{pmatrix}0,0,0,0,0,0,1\ldots\ldots\\0,0,0,1,0,0,0\ldots\ldots\\0,1,0,0,0,0,0\ldots\ldots\\0,0,0,1,0,0,0\ldots\ldots\\0,0,1,0,0,0,0\ldots\ldots\\0,0,0,0,1,0,0\ldots\ldots\\0,1,0,0,0,0,0\ldots\ldots\end{pmatrix}$$

Position Attribute

Due to the different distances from the multi-category word, the POS of the word before or after the multi-category word may, in a POS tagging sequence, have a different influence on the POS tagging of the multi-category word, which we refer to as *position attribute*.

Definition2. *Given a multi-category word with a context window of size l, suppose the number of preceding or following words is n (i.e., l=2n+1), the position attribute vector* V_x *of the multi-category word is given by* $V_x=(d_1,\ldots d_n,d_{n+1},\ldots d_l)$ *where* d_{n+1} *is the value of the position attribute of the multi-category word and* $d_{n+1-i}(d_{n+1+i})$ *is the value of the position attribute of the ith preceding (following) word. We further require that* $d_{n+1-i}=d_{n+1+i}$ *and* $d_{n+1}+\sum_{i=1}^{n}(d_{n+1-i}+d_{n+1+i})=1$.

We choose a proper position attribute vector so that the multi-category word itself has the highest weight, and the closer the surrounding word, the higher its weight is. If we consider a context window of size 7, based on our preliminary experiments, we chose the following position attribute values: $d_1=d_7=1/22$;$d_2=d_6=1/11$;$d_3=d_5=2/11$; and $d_4=4/11$. Hence, the final position attribute vector used in our study can be written as follows:

$$V_X=((1/22),(1/11),(2/11),(4/11),(2/11),(1/11),(1/22)).$$

In the same way, if the size of a context window is 15, the position attribute vector can be described as follows:

$$V_X = ((1/190),(2/190),(4/190),(8/190),(16/190),(32/190),$$
$$((64/190),(32/190),(16/190),(8/190),(4/190),(2/190),(1/190))$$

Note that, if the POS tag in the POS tagging sequence is incorrect, the position attribute value of the corresponding position should be turned into a negative value, so that when the incorrect POS tag appears in a POS tagging sequence, this attribute can correctly show that the incorrect POS tag has had a negative effect on generating the final context vector.

Dependency Attribute

The last attribute we focus on is *dependency attribute*, which corresponds to the fact that there are mutual dependencies on the appearance of every POS in POS tagging sequences. In particular, we use *transition probability* and *emission probability* in Hidden Markov Model (HMM) to capture this dependency.

Definition 3. *Given a tag set of size m(c_1,c_2,…,c_m), the transition probability table T is an m by m matrix and given by:*

$$T_{i,j} = P^T(c_i, c_j) = \frac{f(c_i, c_j)}{f(c_i)}$$

where $f(c_i, c_j)$ is the frequency of the POS tag c_j appears after the POS tag c_i in the entire corpus; $f(c_i)$ is the frequency of the POS tag c_i appearing in the entire corpus; and P^T is the transition probability.

Definition 4. *Given a tag set of size m(c_1,c_2,…,c_m), the emission probability table E is an m by m matrix and given by:*

$$E_{i,j} = P^E(c_i, c_j) = \frac{f(c_i, c_j)}{f(c_j)}$$

where $f(c_i, c_j)$ is the frequency of the POS tag c_i appearing before the POS tag c_j in the entire corpus; $f(c_j)$ is the frequency of POS tag c_j appearing in the entire corpus; and P^E is the emission probability.

Note that both T and E are constructed from the entire corpus and that we can look up these two tables easily when we consider the POS tags appears in POS tagging sequences.

Now, when we look at a context window of size 7 (w_1,w_2,…,w_7) and its POS tagging sequence (t_1,t_2,…,t_7) , there are three types of probabilities we need to take into account.

The first one is the probability of the appearance of the POS tag t_4 of the multi-category word, which we can write as follows:

$$P^{CX}(t_4) = f(\omega_4 \text{ is tagged as } t_4) / f(\omega_4)$$

where $f(w_4)$ is the frequency of the appearance of the multi-category word w_4 in the entire corpus and $f(w_4 \text{ is tagged as } t_4)$ is the frequency of the appearance where the word w_4 is tagged as t_4 in the entire corpus.

The second one is transition probability, which is the probability of the appearance of the POS tag t_{i+1} in the $i+1$ position after the POS tag t_i in the i position and is shown as follows:

$$P^T_{(i,i+1)} = P^T(t_i, t_{i+1}) = f(t_i, t_{i+1})/(t_i)$$

The last is emission probability, which is the probability of the appearance of the POS tag t_{i-1} in the $i-1$ position before the POS tag t_i in the i position and is shown as follows:

$$P^E_{(i-1,i)} = P^T(t_{i-1}, t_i) = f(t_{i-1}, t_i)/(t_i)$$

According to the above three probability formulas, we can build a seven-dimensional vector, where each dimension corresponds to one POS tag, respectively.

Definition 5: *Given a multi-category word with a context window of size 7 and its POS tagging sequence, the dependency attribute vector V_P of the multi-category word is defined as follows:*

$$V_P = (P_1, P_2, P_3, P_4, P_5, P_6, P_7),$$

where

$$P_1 = P^T_{(1,2)} \times P_2;\ P_2 = P^T_{(2,3)} \times P_3;\ P_3 = P^T_{(3,4)} \times P_4;\ P_4 = P^{CX}(t_4);$$

$$P_5 = P^E_{(4,5)} \times P_4;\ P_6 = P^E_{(5,6)} \times P_5;\ P_7 = P^E_{(6,7)} \times P_6;$$

Context Vector of Multi-Category Words

Now we are ready to define the context vector of multi-category words.

Definition 6. *Given a multi-category word with a context window of size l and its POS attribute matrix Y, position attribute vector V_X , and dependency attribute vector V_P, the context vector V_S of the multi-category word is defined as follows:*

$$V_S = (\alpha V_X + \beta V_P) \times Y$$

where α and β are the weights of the position attribute and the dependency attribute, respectively.

Note that we require $\alpha + \beta = 1$, and their optimal values are determined by experiments in our study.

2.2 Experiment on the Size of the Context Window

Context vectors can be extended by using 4 to 7 preceding and following words as a substitute for the 3 preceding and following words we used in context windows and POS tagging sequences. We conducted experiments with a context window of size 3 to 7 on our sampled 1M-word training corpus performing closed tests. The experimental results are simultaneously evaluated in terms of both the precision of consistency checking and algorithm complexity. We plot the effect on precision and relative complexity of the number of preceding or following words in Figure 1 and Figure 2, respectively.

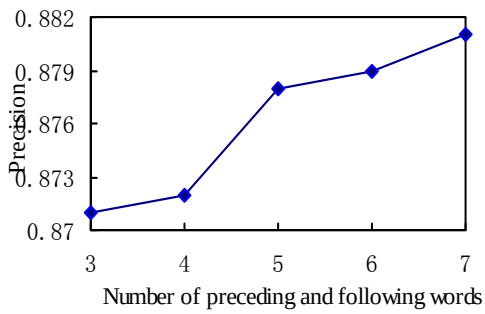

Figure 1. Effect on precision of the number of preceding and following words.

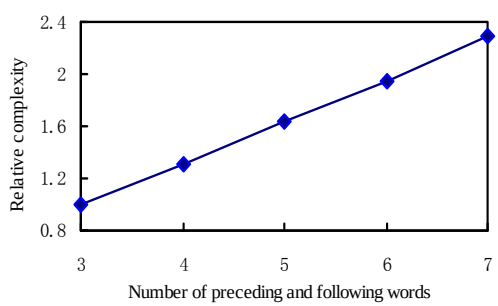

Figure 2. Effect on complexity of the number of preceding and following words.

As shown in Figure 1, the precision of consistency checking increases as we include more preceding and following words. In particular, the precision is improved by 1% when we use 7 preceding and following words.

When the context vector model is extended to contain 7 preceding and following words, the dimensionality of the position attribute vector, POS attribute vector, and dependency

attribute vector increases from 7 to 15, which results in a significant change in run time. Suppose the run time of our algorithm with 3 preceding and following words is 1, then, extending the context vector to 15 dimensions, the change in run time of our algorithm is plotted in Figure 2.

As shown in Figures 1 and 2, the increase of complexity is much higher than that of precision when extending the number of preceding and following words. Hence, we chose 3 as the number of preceding and following words to form context windows and calculate context vectors.

2.3 Effect on Consistency Checking Precision of α and β

When using our sampled 1M-word training corpus to conduct closed test, we found that consistency checking precision changes significantly with different values of α and β. Figure 3 shows the trend when α varies from 0.1 to 0.9.

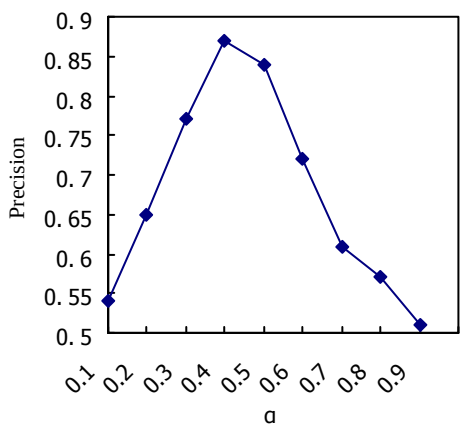

Figure 3. Effect on consistency checking precision of α and β.

As shown in Figure 3, when α=0.4 (β=0.6), consistency checking precision reaches the highest value. Hence, we used α=0.4 and β=0.6 in our experiments. At the same time, the results show α and β have an important effect on consistency checking precision, namely the position attribute vector and the dependency attribute vector are very important for describing the context vector of multi-category word, which is consistent with our experience.

3. Consistency Checking of POS Tagging

Our consistency checking algorithm is based on classification of context vectors of multi-category words. In particular, we first classify context vectors of each multi-category

word in the training corpus, and then we conduct the consistency checking of POS tagging based on classification results.

3.1 Similarity between Context Vectors of Multi-category Words

After constructing context vectors for all multi-category words from their context windows and POS tagging sequences, the similarity of two context vectors is defined as the *Euclidean Distance* between the two vectors.

$$d(x,y) = \|x-y\| = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{1/2}$$

where x and y are two arbitrary context vectors of n dimensions.

3.2 *k*-NN Classification Algorithm

Classification is a process to assign objects that need to be sorted into certain classes. In this paper, we used a popular classification method: the *k*-NN algorithm.

Suppose we have c classes and a class ($\theta_i (i = 1,2,\mathrm{K},c)$) has N_i samples ($x_j^{(i)} (j = 1,2,\ldots N_i)$). The idea of the *k*-NN algorithm is that for each unlabeled object x, compute the distances between x and $M = \sum_{i=1}^{c} N_i$ samples ($x_j^{(i)}$) whose class is known, and select k samples (k nearest neighbors) with the smallest distance. This object x will be assigned to the class that contains the most samples in the k nearest neighbors.

We now formally define the discriminant function and discriminant rule. Suppose $k_1, k_2, \ldots, k_c$ are the numbers of samples in the k nearest neighbors of the object x that belong to the classes $\theta_1, \theta_2, \theta_\mathrm{k}, \theta_c$, respectively. Define the discriminant function of the class θ_i as $d_i(x) = k_i$, $i = 1,2,\ldots,c$. Then, the discriminant rule determining the class of the object x can be defined as follows:

$$d_m(x) = \max_{i=1,2,\mathrm{K},c} d_i(x) \Rightarrow x \in \theta_m$$

3.3 Consistency Checking Algorithm

In this section, we describe the steps of our classification-based consistency checking algorithm in detail.

- Step 1: Randomly sampling sentences containing multi-category words and checking their POS tagging manually. For each multi-category word, classifying the context vectors of the sampled POS tagging sequences, so that the context vectors that have the same POS for the multi-category word belonging to the same class. Let C_{ci} ("*ci*" is a multi-category word) be the number of possible POS tags of the multi-category word *ci*.

- Step 2: Give a context vector *v* of a multi-category word *ci*, calculating the distances between *v* and all the context vectors that contain the multi-category word *ci* in the training corpus, and selecting *k* context vectors whose distances are smaller than the others.
- Step 3: According to the k-NN algorithm, checking the classes of the k nearest context vectors and classifying the vector *v*.
- Step 4: Comparing the POS of the multi-category word *ci* in the class that the k-NN algorithm assigns *v* to and the POS tag of *ci*,. If they are the same, the POS tagging of the multi-category word *ci* is considered to be consistent, otherwise it is inconsistent.

The major disadvantage of this algorithm is the difficulty in selecting the value of *k*. If *k* is too small, the classification result is unstable; on the other hand, if *k* is too big, the classification deviation increases. Therefore, the selection of the value of *k* should depend on not only some general factors, but also the unique characteristics and structures of the data set. Hence, the best method of selecting *k* should be a strategy that has an ability to adapt different data sets. One such strategy is to try different values on the training set first, draw the performance curve, and select an optimal *k* value.

3.4 Selecting *k* in Classification Algorithm

Figure 4 shows the consistency checking precision values obtained with various *k* values in the *k*-NN algorithm. The precision values are closed test results on our 1M-word training corpus, and were obtained by using α =0.4 and β =0.6 in the context vector model.

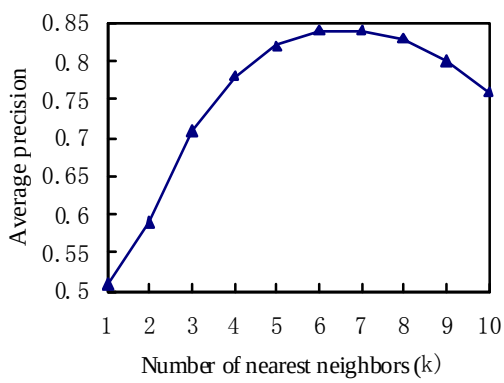

Figure 4. Effect on precision of k in the k-NN algorithm.

As shown in Figure 4, when *k* continues to increase from 6, the precision remains the same. When *k* reaches 9, the precision starts declining. Our experiment with other α and β values also show similar trends. Hence, we chose *k*=6 in this paper.

4. Experimental Results and Discussions

We evaluated our consistency checking algorithm on our 1.5M-word corpus (including the 1M-word training corpus) and conducted open and closed tests. The results are showed in Table 1.

Table 1. Experimental Results.

Test corpora	Test type	Number of multi-category words	Number of the true inconsistencies	Number of the identified inconsistencies (True)	Recall (%)	Precision (%)
1M word	closed	127,210	1,147	1,219(1063)	92.67	87.20
500K word	open	64,467	579	583(497)	85.84	85.24

$$\text{Recall} = \frac{\text{Number of the identified true inconsistencies}}{\text{Number of the true inconsistencies}}$$

$$\text{Precision} = \frac{\text{Number of the identified true inconsistencies}}{\text{Number of the identified inconsistencies}}$$

The experimental results show two interesting trends. First, the precision and recall of our consistency checking algorithm are 87.20% and 92.67% in the closed test, respectively, and 85.24% and 85.84% in the open test, respectively. Compared to Zhang *et al.* 2004, the precision of consistency checking is improved by 2~3%, and the recall is improved by 10%. The experimental results indicate that the context vector model has great improvements over the one used in Zhang *et al*. 2004. Second, thanks to the considerable improvement of the recall, to some extent, our consistency checking algorithm prevents the occurrence of events with small probabilities in POS tagging.

5. Conclusion and Future Research

After analyzing the experimental results we have come to the following conclusions:

- The proposed classification-based method is effective for consistency checking of POS tagging. It solves a difficult problem that easily appears in automatic POS tagging processes, and greatly improves the efficiency and quality of manual correction of automatic POS tagging.
- The information in corpora can be considered as if it were in a multi-dimensional space with various attributes that have mutual influences. When conducting a consistency checking of POS tagging, it is incomplete if we only consider POS tagging and relationships between the POS without taking into account other attributes of words. We

may use other widely known attributes of words to improve the consistency checking of POS tagging.

In the future, we plan to investigate more types of word attributes and incorporate linguistic and mathematical knowledge to develop better consistency checking models, which ultimately provide a better means of building high-quality Chinese corpora.

Acknowledgements

This research was partially supported by the National Natural Science Foundation of China No. 60473139, 60775041 and the Natural Science Foundation of Shanxi Province No. 20051034.

References

Du, Y.-P., and J.-H. Zheng, "The proofreading method study on consistence of segment and part-of-speech," *Computer Development and Application*, 14(10), 2001, pp. 16-18.

Huang, C.-R., and R.-Y. Chang, "Categorical Am biguity and Information Content: A Corpus-based Study of Chinese," *Journal of Chinese Language and Computing*, 14(2), 2004, pp. 157-165.

Ng, H. T., and K. L. Jin, "Chinese Part-of-Speech Tagging: One-at-a-Time or All-at-Once? Word-Based or Character-Based?," *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, 2004, Barcelona, Spain, pp. 277-284.

Qian, W.-N., and A.-Y. Zhou, "Analyzing Popular Clustering Algorithms from Different Viewpoints," *Journal of Software*, 13(8), 2002, pp. 1382-1394.

Qian, Y.-L., and J.-H. Zheng, "Research on the method of automatic correction of chinese POS tagging," *Journal Of Chinese Information Processing*, 18(2), 2003, pp. 30-35.

Qian, Y.-L., and J.-H. Zheng, "An approach to improving the quality of part-of-speech tagging of Chinese text," *In Proceedings of the 2004 IEEE International Conference on Information Technology:Coding and Computing (ITCC 2004)*, 2004, USA, pp. 183-187.

Qu, W.-G., and X.-H. Chen, "Research and practice on the machine proofreading of the tagged corpora," *In Proceeding of the 7th National Computation Linguistics (JSCL'03)*, 2003, Beijing, China, pp. 318-324.

Xing, H.-B., "Analysing on the words classified hard in pos tagging," *In Proceedings of the 5th National Computational Linguistics (JSCL'99)*, 1999, Beijing, China, pp. 187-192.

Xue, N., F.-D. Chiou, and M. Palmer, "Building a Large-Scale Annotated Chinese Corpus," *In Proceedings of the 19th international conference on Computational linguistics*, 2002, Taipei, Taiwan, pp. 1-8.

Zhang, H., and J.-H. Zheng, "The Inspecting Method Study on Consistency of POS Tagging of Corpus," *Journal Of Chinese Information Processing*, 18(5), 2004, pp. 11-16.

Zhang, M., and S. Li, "POS Tagging Chinese Corpus Based on Statistics and Rules," *Journal of Software*, 9(2), 1998, pp. 134-138.