# A Classification-based Algorithm for Consistency Check of Part-of-Speech Tagging for Chinese Corpora

**Hu Zhang** **Jia-heng Zheng**
School of Computer
& Information Technology
Shanxi University
Taiyuan, Shanxi 030006, China

**Ying Zhao** *
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455, USA
yzhao@cs.umn.edu

## Abstract

Ensuring consistency of Part-of-Speech (POS) tagging plays an important role in constructing high-quality Chinese corpora. After analyzing the POS tagging of multi-category words in large-scale corpora, we propose a novel consistency check method of POS tagging in this paper. Our method builds a vector model of the context of multi-category words, and uses the $k$-NN algorithm to classify context vectors constructed from POS tagging sequences and judge their consistency. The experimental results indicate that the proposed method is feasible and effective.

## 1 Introduction

Constructing high-quality and large-scale corpora has always been a fundamental research area in the field of Chinese natural language processing. In recent years, the rapid development in the fields of machine translation (MT), phonetic recognition (PR), information retrieval (IR), web text mining, and etc., is demanding more Chinese corpora of higher quality and larger scale. Ensuring consistency of Part-of-Speech (POS) tagging plays an important role in constructing high-quality Chinese corpora. In particular, we focus on consistency check of the POS tagging of multi-tagging words, which consist of same Chinese characters and are near-synonymous, but have different grammatical functions. No matter how many different POS tags a multi-category words may be tagged, ensuring consistency of POS tagging means to assign the multi-category word with the same POS tag when it appears in similar context.

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 89% and 96%, respectively. A great portion of the words appearing in Chinese corpora are multi-category words. We have studied the text data from the 2M-word Chinese corpus published by Peking University, and statistics show that multi-category words cover 11% of the words, while the percentage of the occurrence of multi-category words is as high as 47%. When checking the POS tags, human experts may have disagreements or make mistakes in some cases. After analyzing 1,042 sentences containing the word "高", which are extracted from the 2M-word Chinese corpus of Peking University, the number of incorrect tags for the word "高" is 15, which accounts for around 1.3%.

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 has focused on consistency check of POS tagging. Xing (Xing, 1999) analyzed the inconsistency phenomena of word segmentation (WS) and POS tagging. Qu and Chen (Qu and Chen, 2003) improved the corpus quality by obtaining POS tagging knowledge from processed corpora, preprocessing, and checking con-

*To whom correspondence should be addressed.

sistency with methods based on rules and statistics. Qian and Zheng (Qian and Zheng, 2003; Qian and Zheng, 2004) introduced a rule-based consistency check 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 (Du and Zheng, 2001) proposed a rule-based consistency check method and strategy to identify the inconsistency phenomena of POS tagging. However, the algorithms and techniques for automatic consistency check of POS tagging proposed in (Qu and Chen, 2003; Qian and Zheng, 2003; Qian and Zheng, 2004; Du and Zheng, 2001) 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. (Zhang et al., 2004), the proposed method fully considers the mutual relation of the POS in POS tagging sequence, and adopts transition probability and emission probability to describe the mutual dependencies and $k$-NN algorithm to weigh the similarity. We evaluated our proposed algorithm on our 1.5M-word corpus. In open test, our method achieved a precision of 85.24% and a recall 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 check 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 context 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 multi-category word and the tagging sequence of this window. In the rest of this section, we describe our 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 $(w_1, w_2, ..., w_l)$ and the POS tags of each word as $(t_1, t_2, ..., t_l)$. We also refer to the latter vector as *POS tagging sequence*.

In practise, we choose a proper value of $l$ so that the context window contains 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.

#### 2.1.1 POS Attribute

The POS tagging sequence contains information of the POS of each preceding (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.

Suppose we have a tag set of size $m$ $(c_1, c_2, ..., c_m)$, given a multi-category word with a context window of size $l$ $(w_1, w_2, ..., w_l)$ and its POS tagging sequence, the *POS attribute matrix* $Y$ is an $l$ by $m$ matrix, where the rows indicate the POS tags of the preceding words, 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 $w_i$ is $c_j$.

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

"高" 缀/v 满/a 彩灯/n 的/u 高/a 塔/n 直/d 插/v 夜空/n ,/w

As we let $l = 7$, we look at the word "高" and its 3 preceding and following words. Hence, the POS tagging sequence is ( a, n, u, a, n, d, 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 "高" in this example is:

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

#### 2.1.2 Position Attribute

Due to the different distances from the multi-category word, the POS of the word before (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*.

Given a multi-category word with a context window of size $l$, suppose the number of preceding (following) words is $n$ (i.e., $l = 2n + 1$), the *position attribute vector* $V_X$ of the multi-category word is given by $V_X = (d_1, ..., d_n, d_{n+1}, d_{n+2}, ..., d_l)$, 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 $i$th preceding (following) word. We further require that $\forall i$ $d_{n+1-i} = d_{n+1+i}$ 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:

$$V_X = (\frac{1}{22}, \frac{1}{11}, \frac{2}{11}, \frac{4}{11}, \frac{2}{11}, \frac{1}{11}, \frac{2}{22}).$$

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 negative effect on generating the final context vector.

#### 2.1.3 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) (Leek, 1997) to capture this dependency.

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 $c_i$ appears in the entire corpus; and $P^T$ is the *transition probability*.

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

Note that both $T$ and $E$ are constructed from the entire corpus and we can look up these two tables easily when we consider the POS tags appear 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:

$$P^{CX}(t_4) = f(w_4 \text{ is tagged as } t_4)/f(w_4), \quad (1)$$

where $f(w_4)$ is the frequency of the appearance of the multi-category word $w_4$ in the entire corpus and $f(w_4 istaggedast_4)$ 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 shown in Eqn. 2:

$$P^T_{(i,i+1)} = P^T(t_i, t_{i+1}) = f(t_i, t_{i+1})/f(t_i). \quad (2)$$

The last 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 shown in Eqn. 3:

$$P^E_{(i-1,i)} = P^E(t_{i-1}, t_i) = f(t_{i-1}, t_i)/f(t_i). \quad (3)$$

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

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

$$\begin{aligned}
P_1 &= P^T_{(1,2)} \cdot P_2 \\
&= P^T_{(1,2)} \cdot P^T_{2,3} \cdot P^T_{(3,4)} \cdot P^{CX}(t_4); \\
P_2 &= P^T_{(2,3)} \cdot P_3 = P^T_{(2,3)} \cdot P^T_{(3,4)} \cdot P^{CX}(t_4); \\
P_3 &= P^T_{(3,4)} \cdot P_4 = P^T_{(3,4)} \cdot P^{CX}(t_4); \\
P_4 &= P^{CX}(t_4); \\
P_5 &= P^T_{(4,5)} \cdot P_4 = P^E_{(4,5)} \cdot P^E_{(4,5)} \cdot P^{CX}(t_4); \\
P_6 &= P^T_{(5,6)} \cdot P_5 = P^E_{(5,6)} \cdot P^E_{(4,5)} \cdot P^{CX}(t_4); \\
P_7 &= P^T_{(6,7)} \cdot P_6 \\
&= P^E_{(6,7)} \cdot P^E_{(5,6)} \cdot P_{(4,5)} \cdot P^{CX}(t_4).
\end{aligned}$$

#### 2.1.4 Context Vector of Multi-category Words

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

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:

$$V_S = (\alpha V_X + \beta V_P) \times Y, \quad (4)$$

where $\alpha$ and $\beta$ 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 (following) words to substitute 3 preceding (following) words 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 and performed closed test. The experimental results are evaluated in terms of both the precision of consistency check and algorithm complexity simultaneously. We plot the effect on precision in Figure 1.

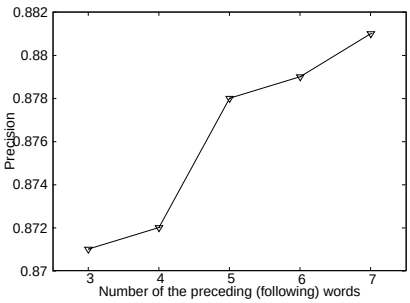


Figure 1: Effect on precision of the number of preceding (following) words.

As shown in Figure 1, the precision of consistency check increases as we include more preceding (following) words. In particular, the precision is improved by 1% when we use 7 preceding (following) words. However, the increase of complexity is much higher than that of precision, because the dimensionality of the position attribute vector, POS attribute vector, and dependency attribute vector doubles. Hence, we chose 3 as the number of preceding (following) words to form context windows and calculate context vectors.

### 2.3 Effect on consistency check precision of $\alpha$ and $\beta$

When using our sampled 1M-word training corpus to conduct closed test, we found that consistency check precision changes significantly with the different values of $\alpha$ and $\beta$. Figure 2 shows the trend when $\alpha$ varies from 0.1 to 0.9. We used $\alpha = 0.4$ and $\beta = 0.6$ in our experiments.

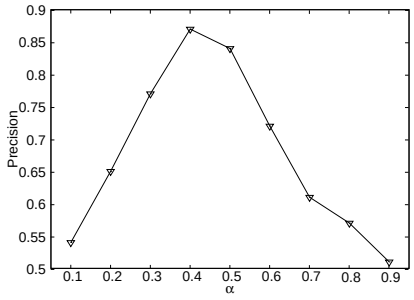


Figure 2: Effect on consistency check precision of $\alpha$ and $\beta$.

## 3 Consistency Check of POS Tagging

Our consistency check 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 check of POS tagging based on classification results.

### 3.1 Similarity between Context Cectors 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.

$$d(x,y) = \|x-y\| = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{(1/2)}, \quad (5)$$

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 classified to a certain class. In this paper, we used a popular classification method: the $k$-NN algorithm.

Suppose we have $c$ classes and a class ($\omega_i (i = 1, 2, ..., c)$) has $N_i$ samples ($x_j^{(i)} (j = 1, 2, ..., N_i)$). The idea of the $k$-NN algorithm is that for each unlabeled object $x$, compute the distances between $x$ and all samples 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 $k_1, k_2, ..., k_c$ are the numbers of samples in the $k$ nearest neighbors of the object $x$ that belong to the classes $\omega_1, \omega_2, ..., \omega_c$, respectively. Define the discriminant function of the class $\omega_i$ as $d_i(x) = k_i, i = 1, 2, ..., c$. Then, the discriminant rule of determining the class of the object $x$ can be defined as follows:

$$d_m x = \max_{i=1,2,...,c} d_i(x) \Rightarrow x \in \omega_m$$

### 3.3 Consistency Check Algorithm

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

**Step1:** 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 belong to the same class.

**Step2:** Given a context vector $x$ of a multi-category word $c$, calculating the distances between $x$ and all the context vectors that contains the multi-category word $c$ in the training corpus, and selecting $k$ context vectors with smallest distances.

**Step3:** According to the $k$-NN algorithm, checking the classes of the $k$ nearest context vectors and classifying the vector $x$.

**Step4:** Comparing the POS of the multi-category word $c$ in the class that the $k$-NN algorithm assigns $x$ to and the POS tag of $c$. If they are the same, the POS tagging of the multi-category word $c$ 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.

### 3.4 Selecting $k$ in Classification Algorithm

Figure 3 shows the consistency check precision values obtained with various $k$ values in the $k$-NN algorithm. The precision values are closed

Table 1: Experimental Results

| Test corpora | Test type | Number of multi-category words | Number of the true inconsistencies | Number of the identified inconsistencies | Recall (%) | Precision (%) |
|---|---|---|---|---|---|---|
| 1M-word | closed | 127,210 | 1,147 | 1,219 (156) | 92.67 | 87.20 |
| 500K-word | open | 64,467 | 579 | 583 (86) | 85.84 | 85.24 |

test results on our 1M-word training corpus, and were obtained by using $\alpha = 0.4$ and $\beta = 0.6$ in the context vector model.

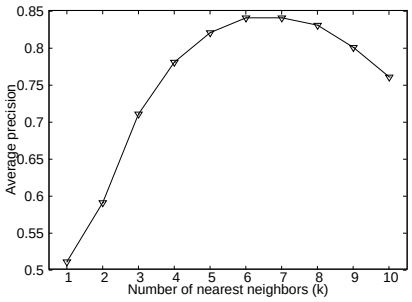


Figure 3: Effect on precision of $k$ in the $k$-NN algorithm.

As shown in Figure 3, when $k$ continues to increase from 6, the precision remains the same. When when $k$ reaches to 9, the precision starts declining. Our experiment with other $\alpha$ and $\beta$ values also show similar trends. Hence, we chose $k = 6$ in this paper.

## 4 Experimental Results

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

The experimental results show two interesting trends. First, the precision and recall of our consistency check algorithm are 87.20% and 92.67% in closed test, respectively, and 85.24% and 85.84% in open test, respectively. Compared to Zhang et al. (Zhang et al., 2004), the precision of consistency check is improved by 2$\sim$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. (Zhang et al., 2004). Second, thanks to the great improvement of the recall, to some extent, our consistency check algorithm prevents the happening of events with small probabilities in POS tagging.

## 5 Conclusion and Future Research

In this paper, we propose a new classification-based method to check consistency of POS tagging, and evaluated our method on our 1.5M-word corpus (including 1M-word training corpus) with both open and closed tests.

In the future, we plan to investigate more types of word attributes and incorporate linguistic and mathematical knowledge to develop better consistency check 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 and the Natural Science Foundation of Shanxi Province No. 20051034.

## References

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

T. R. Leek. 1997. Information extraction using hidden Markov models. Master's thesis, UC San Diego.

Y. Qian and J. Zheng. 2003. Research on the method of automatic correction of chinese pos tagging. *Journal of Chinese Information Processing*, 18(2):30–35.

Y. Qian and J. Zheng. 2004. 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)*.

W. Qu and X. Chen. 2003. Analysing on the words classified hard in pos tagging. In *Proceedings of the 9th National Computational Linguistics (JSCL'03)*.

H. Xing. 1999. Analysing on the words classified hard in pos tagging. In *Proceedings of the 5th National Computational Linguistics (JSCL'99)*.

H. Zhang, J. Zheng, and J. Liu. 2004. The inspecting method study on consistency of pos tagging of corpus. *Journal of Chinese Information Processing*, 18(5):11–16.