KCAT : A Korean Corpus Annotating Tool Minimizing Human Intervention

Won-Ho Ryu, Jin-Dong Kim, Hae-Chang Rim
Dept. of Computer Science & Engineering,
Natural Language Processing Lab,
Korea University
Anam-dong 5-ga, Seongbuk-gu, Seoul, Korea
whryu, jin, rim @nlp.korea.ac.kr

Heui-Seok Lim
Information Communications Department,
Natural Language Processing Lab,
Chonan University
85-1, Anseo-Dong, Chonan City,
ChungChong-NamDo Province, Korea
limhs@infocom.chonan.ac.kr

Abstract

While large POS(part-of-speech) annotated corpora play an important role in natural language processing, the annotated corpus requires very high accuracy and consistency. To build such an accurate and consistent corpus, we often use a manual tagging method. But the manual tagging is very labor intensive and expensive. Furthermore, it is not easy to get consistent results from the human experts. In this paper, we present an efficient tool for building large accurate and consistent corpora with minimal human labor. The proposed tool supports semi-automatic tagging. Using disambiguation rules acquired from human experts, it minimizes the human intervention in both the manual tagging and post-editing steps.

1. Introduction

The POS annotated corpora are very important as a resource of useful information for natural language processing. A problem for corpus annotation is the trade-off between efficiency and accuracy.

Although manual POS tagging is very reliable, it is labor intensive and hard to make a consistent POS tagged corpus. On the other hand, automatic tagging is prone to errors for infrequently occurring words due to the lack of overall linguistic information. At present, it is almost impossible to construct a highly accurate corpus by using an automatic tagger alone.

As a consequence, a semi-automatic tagging method is proposed for corpus annotation. In ordinary semi-automatic tagging, an automatic tagger tags each word and human experts correct the mis-tagged words in the post-editing step. But, in the post-editing step, as the human expert cannot know which word has been annotated incorrectly, he must check every word in the whole corpus. And he must do the same work again and again for the same words in the same context. This situation causes as much labor-intensive work as in manual tagging.

In this paper, we propose a semi-automatic tagging method that can reduce the human labor and guarantee the consistent tagging.

2. System Requirements

To develop an efficient tool that attempts to build a large accurately annotated corpus with minimal human labor, we must consider the following requirements:

- In order to minimize human labor, the same human intervention to tag and to correct the same word in the same context should not be repeated.
- There may be a word which was tagged inconsistently in the same context because it was tagged by different human experts or at a different task time. As an efficient tool, it can prevent the inconsistency of the annotated results and guarantee the consistency of the annotated results.
- It must provide an effective annotating capability for many unknown words in the whole corpus.

3. Proposed POS Tagging Tool:KCAT

The proposed POS tagging tool is used to combine the manual tagging method and the automatic tagging method. They are integrated to increase the accuracy of the automatic tagging method and to minimize the amount of the human labor of the manual tagging method. Figure 1 shows the overall architecture of the proposed tagging tool :KCAT.

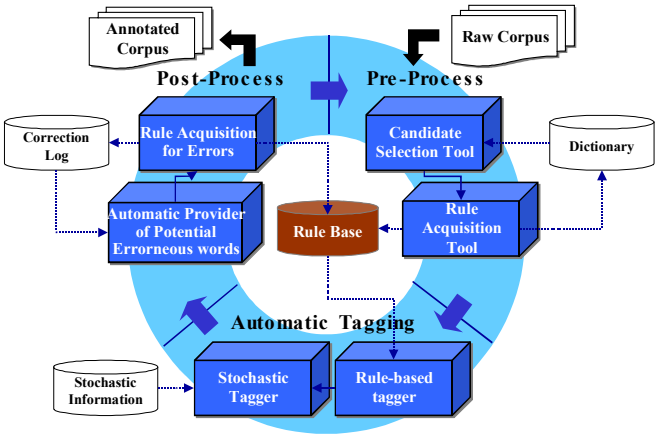

Figure 1. System Architecture of KCAT

As shown in figure 1, KCAT consists of three modules: the pre-processing module, the automatic tagging module, and the post-processing module. In the pre-processing module, the disambiguation rules are acquired from human experts. The candidate words are the target words whose disambiguation rules are acquired. The candidate words can be unknown words and also very frequent words. In addition, the words with problematic ambiguity for the automatic tagger can become candidates. Disambiguation rules are acquired with minimal human labor using the tool proposed in (Lee,1996). In the automatic tagging module, the disambiguation rules resolve the ambiguity of every word to which they can be applied. However, the rules are certainly not sufficient to resolve all the ambiguity of the whole words in the corpus. The proper tags are assigned to the remaining ambiguous words by a stochastic tagger. After the automatic tagging, a human expert corrects the errors of the stochastic tagger. The system presents the expert with the results of the stochastic tagger. If the result is incorrect, the human expert corrects the error and generates a disambiguation rule for the word. The rule is also saved in the rule base in order to be used later.

3.1. Lexical Rules for Disambiguation

There are many ambiguous words that are extremely difficult to resolve ambiguities by using a stochastic tagger. Due to the problematic words, manual tagging and manual correction must be done to build a correct corpus. Such human intervention may be repeated again and again to tag or to correct the same word in the same context.

For example, a human expert should assign '*Nal*(flying)/Verb+*Neun*/Ending' to every '*NaNeun*' repeatedly in the following sentences:

"*Keu-Nyeo-Neun Ha-Neul-Eul* ***Na-Neun*** *Pi-Haeng-Ki-Reul Pon Ceok-i Iss-Ta.*" (she has seen a flying plane)
"*Keu-Neun Ha-Neul-Eul* ***Na-Neun*** *Pi-Haeng-Ki-Reul Pon Ceok-i Eops-Ta.*" (he has never seen a flying plane)
"*Keu-Neun Ha-Neul-Eul* ***Na-Neun*** *Pi-Haeng-Ki-Reul Pal-Myeong-Haess-Ta.*" (he invented a flying plane)
:

In the above sentences, human experts can resolve the word, '*Na-Neun*' with only the previous and the next lexical information: '*Ha-Neul-Eul*' and '*Pi-Haeng- Ki-Reul*'. In other words, the human expert has to waste time on tagging the same word in the same context repeatedly. This inefficiency can also be happened in the manual correction of the mis-tagged words. So, if the human expert can make a rule with his disambiguation knowledge and use it for the same words in the same context, such inefficiency can be minimized. We define the disambiguation rule as a lexical rule. Its template is as follows.

[P:N] [Current Word] [Context] = [Tagging Result]

Context : Previous words$^{0}_{p}$ * Next Words$^{0}_{n}$

In the above template, *p* and *n* mean the previous and the next context size respectively. For the present, *p* and *n* are limited to 3. '*'

represents the separating mark between the previous and next context. For example, the rule [1:1] [*Na-Nuen*] [*Ha-Neul-Eul* * Pi-Haeng-Ki-Reul] = [*Nal*(flying)/Verb + *Neun*/Ending] says the tag '*Nal*(flying)/Verb + *Neun*/Ending' should be assigned to the word '*Na-Neun*' when the previous word and the next word is '*Ha-Neul-Eul*' and '*Pi-Haeng-Ki-Reul*'.

Although these lexical rules cannot always correctly disambiguate all Korean words, they are enough to cover many problematic ambiguous words. We can gain some advantages of using the lexical rule. First, it is very accurate because it refers to the very specific lexical information. Second, the possibility of rule conflict is very little even though the number of the rules is increased. Third, it can resolve problematic ambiguity that cannot be resolved without semantic information(Lim,1996).

3.2. Lexical Rule Acquisition

Lexical rules are acquired for the unknown words and the problematic words that are likely to be tagged erroneously by an automatic tagger. Lexical rule acquisition is performed by following steps:

1. The system builds a candidate list of words for which the lexical rules would be acquired. The candidate list is the collection of all examples of unknown words and problematic words for an automatic tagger.
2. A human expert selects a word from the list and makes a lexical rule for the word.
3. The system applies the lexical rule to all examples of the selected word with same context and also saves the lexical rule in the rule base.
4. Repeat the steps 2 and 3 until all examples of the candidate words can be tagged by the acquired lexical rules.

3.3. Automatic Tagging

In the automatic tagging phase, words are disambiguated by using the lexical rules and a stochastic tagger. To annotate a word in a raw corpus, the rule-based tagger first searches the lexical rule base to find a lexical rule that can be matched with the given context. If a matching rule is found, the system assigns the result of the rule to the word. According to the corresponding rule, a proper tag is assigned to a word. With the lexical rules, a very precise tag can be assigned to a word. However, because the lexical rules do not resolve all the ambiguity of the whole corpus, we must make use of a stochastic tagger. We employ an HMM-based POS tagger for this purpose(Kim,1998). The stochastic tagger assigns the proper tags to the ambiguous words after the rule application.

After disambiguating the raw corpus using the lexical rules and the automatic tagger, we arrive at the fully disambiguated result. But the word tagged by the stochastic tagger may have a chance to be mis-tagged. Therefore, the post-processing for error correction is required for the words tagged by the stochastic tagger.

3.4. Error Correction

The human expert carries out the error correction task for the words tagged by a stochastic tagger. This error correction also requires the repeated human labor as in the manual tagging. We employ the similar way of the rule acquisition to reduce the human labor needed for manual error correction. The results of the automatic tagger are marked to be distinguished from the results of the rule-based tagger. The human expert checks the marked words only. If an error is found, the human expert assigns a correct tag to the word. When the expert corrects the erroneous word, the system automatically generates a lexical rule and stores it in the rule base. The newly acquired rule is automatically applied to the rest of the corpus. Thus, the expert does not need to correct the repeated errors.

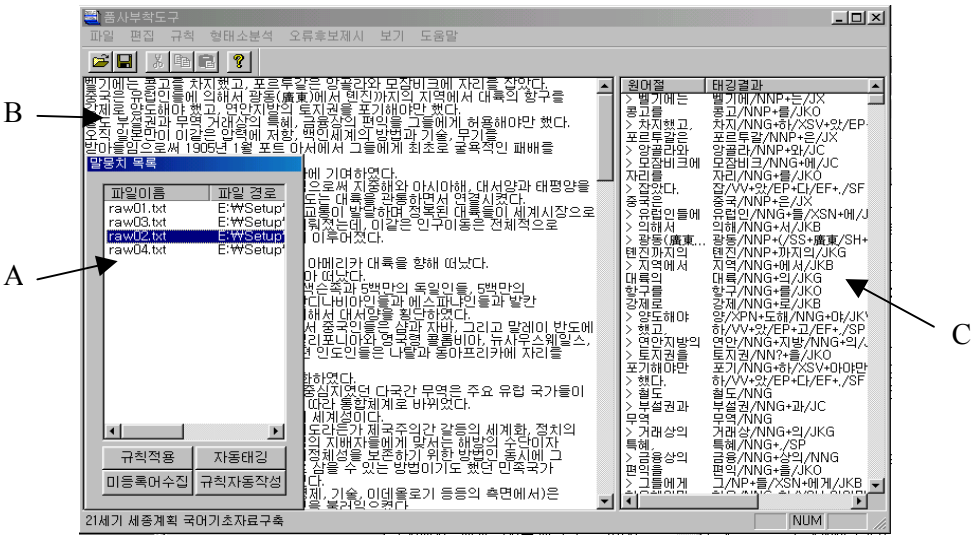

Figure 2. Building Annotated Corpus Using KCAT

4. Application to Build Large Corpora

Based on the proposed method, we have implemented a corpus-annotating tool for Korean which is named as KCAT(Korean Corpus Annotating Tool). The process of building large corpora with KCAT is as follows:

1. The lexical rules in the rule base are applied to a raw corpus. If the rule base is empty, nothing will be done.
2. The system makes a candidate list.
3. Human expert produces the lexical rules for the words in the candidate list.
4. The system tags the corpus by using the lexical rules and a stochastic tagger.
5. Human manually corrects errors caused by the stochastic tagger, and lexical rules for those errors are also stored in the rule-base.
6. For other corpus, repeat the steps 1 through 5.

Figure 2 shows a screenshot of KCAT. In this figure, 'A' window represents the list of raw corpus and a 'B' window contains the content of the selected raw corpus in the window A. The tagging result is displayed in the window 'C'. Words beginning with '>' are tagged by a stochastic tagger and the other words are tagged by lexical rules.

We can get the more lexical rules as the tagging process is progressed. Therefore, we can expect that the accuracy and the reduction rate of human labor are increased as long as the tagging process is continued.

5. Experimental Results

In order to estimate the experimental results of our system, we collected the highly ambiguous words and frequently occurring words in our test corpus with 50,004 words. Table 1 shows reductions in human intervention required to annotate the raw corpus when we use lexical rules for the highly ambiguous words and the frequently occurring words respectively. The second column shows that we examined the 4,081 occurrences of 2,088 words with tag choices above 7 and produced 4,081 lexical rules covering 4,832 occurrences of the corpus. In this case, the reduction rate of human intervention is 1.5%.[1] The third column shows that we examined the 6,845 occurrences of 511 words with frequency above 10 and produced 6,845 lexical rules covering 15,418 occurrences of the corpus. In this case, the reduction rate of human intervention is 17%.[2]

The last row in the table shows how informative the rules are. We measured it by the improvement rate of stochastic tagging after the rules are applied. From these experimental results, we can judge that rule-acquisition from frequently occurring words is preferable.

---

[1] (4,832-4,081) / 50,004

[2] (15,418-6,845) / 50,004

Table 1. Reduction in human Intervention

Type of word for rule acquisition	Ambiguous words (≥7)	Frequently occurring words (≥10)
Number of words	4832(9.6%)	15418(30%)
Number of lexical rules	4081	6845
Decrement of human intervention	1.5%	17%
Improvement of tagging accuracy	1.6% (94.1-92.5%)	3.7% (95.2-92.5%)

Table 2 shows the results of our experiments on the applicability of lexical rules. We measure it by the improvement rate of stochastic tagging after the rules acquired from other corpus are applied.

The third row shows that we annotate a training corpus with 10,032 words and produce 631 lexical rules, which can be applied to another test corpus to reduce the number of the stochastic tagging errors from 697 to 623.[3]

The forth and fifth row show that as the number of lexical rules is increased, the number of the errors of the tagger is decreased on the test corpus.

These experimental results demonstrate the promise of gradual decrement of human intervention and improvement of tagging accuracy in annotating corpora.

Table 2. Applicability of Lexical Rules

Size of the corpus	The number of lexical rules	The number of stochastic errors
0	0	697
10,032	631	623
20,047	1361	565
30,049	2091	538

6. Conclusion

The main goal of our work is to develop an efficient tool which supports to build a very accurately and consistently POS annotated corpus with minimal human labor. To achieve the goal, we have proposed a POS tagging tool named KCAT which can use human linguistic knowledge as a lexical rule form. Once a lexical rule is acquired, the human expert doesn't need to spend time in tagging the same word in the same context. By using the lexical rules, we could have very accurate and consistent results as well as reducing the amount of the human labor.

It is obvious that the more lexical rules the tool acquires the higher accuracy and consistency it achieves. But it still requires a lot of human labor and cost to acquire many lexical rules. And, as the number of the lexical rules is increased, the speed of rule application is decreased. To overcome the barriers, we try to find a way of rule generalization and a more efficient way of rule encoding scheme like the finite-state automata(Roche,1995).

Furthermore, we will use the distance of the best and second tag's probabilities to classify reliable automatic tagging result and unreliable tagging result(Brants,1999).

[3] Our test corpus includes 10,015 words

References

Brants, T. Skut, W. and Uszkoreit, H. (1999) *Syntactic Annotation of a German Newspaper Corpus*. In "Journees ATALA", pp.69-76.

Kim, J. D. Lim, H. S. and Rim, H. C. (1998) *Morpheme-Unit POS Tagging Model Considering Eojeol-Spacing*. In "Proc. of the 10th Hangul and Korean Information Processing Conference", pp.3-8.

Lee, J. K. (1996) *Eojeol-unit rule Based POS tagging with minimal human intervention.* M. S dissertation, Dept. of Computer Science and Engineering, Korea Univ.

Lim, H. S. Kim, J. D. and Rim, H. C. (1996) *A Korean Transformation-based POS Tagger with Lexical Information of mistagged Eojeol* . In "Proc. of the 2nd Korea-China Joint Symposium on Oriental Language Computing", pp.119-124.

Roche, E. and Schabes, Y. (1995) *Deterministic Part-of-Speech Tagging with Finite-State Transducer*. Computational Linguistics, 21/2, pp. 227-253.