Automatic Preposition Errors Correction Using Inductive Learning

Hokuto Ototake Kenji Araki

Graduate School of Information Science and Technology
Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo, 060-0814 Japan
E-mail: {hokuto, araki}@media.eng.hokudai.ac.jp

Abstract

In this paper, we describe a system for correcting English preposition errors automatically. Non-native English writers often make these errors. Our system uses rules extracted automatically based on preposition context features, such as preceding and following nouns. Additional rules are generated recursively from the extracted rules using Inductive Learning. Our system achieves 82% accuracy and 32% coverage, which are competitive with other systems. Apart from the performance, it has an advantage of being more understandable while investigating why a given preposition was erroneous. This is because we use rules and they give this advantage over maximum entropy approaches.

1. Introduction

It is one of the difficult problems for learners of English to use prepositions properly. Izumi et al. [5] reported error rates for English prepositions that were as high as 10% in a Japanese learners corpus. Felice et al. [2] also reported that 12% of errors were prepositions in a small error-tagged corpus they created.

Consequently, a teacher of English has to correct a lot of preposition errors in writing. In many circumstances, they need to look up in a dictionary or check real text examples when correcting these errors because of the great variety of linguistic functions that prepositions serve.

In view of these circumstances, methods for detecting and correcting preposition errors automatically have been proposed in the past. Izumi et al. [4] used a maximum entropy approach to recognize various errors using contextual features. They do not show performance of prepositions specifically, but overall performance of the targeted 13 error types achieved 25% precision and 7% recall. Gamon et al. [3] proposed a complex system including a language model and decision trees to detect preposition and determiner errors. Their system performed at 79% precision, but recall was not shown. Tetreault et al. [8] used a maximum entropy classifier to build a model of correct preposition usage for 34 common English prepositions. They reported 84% precision and 19% recall. And, Felice et al. [2] used a maximum entropy approach to correct preposition and determiner errors. They reported 70% accuracy of preposition classifying in native English writing.

In this paper, we propose a system for correcting preposition errors automatically using Inductive Learning (IL) [1] which we developed originally. The system is based on the error correction algorithm [7] that we previously proposed for determiner errors, which uses rules involving a combination of a preposition and feature slots constructed from preceding and following context. Rules are extracted from corpora of error free text. Also, additional abstract rules are generated recursively from the extracted rules by IL. This has the advantage of making rules automatically and finding context factors relating to preposition selection by abstracting rules. And we believe that our system is more understandable about the reason why a preposition is incorrect than others based on a maximum entropy approach, because of using the rules.

The rest of the paper is organized as follows: In section 2, we describe the system for correcting preposition errors. In section 3, we evaluate the system and section 4 includes our conclusions.

2. System

Figure 1 shows the process flow. When the system gets an English sentence including preposition errors as user input, it first parses the sentence with The Stanford Parser [6] and extracts feature slots. Using the resultant feature slots, the system extracts rules for error correction from a rule database that is constructed from a large corpus in advance. From the extracted rules, the system generates new abstract rules by IL. And the system calculates scores of rules to rank them by reliability. Finally, the rules are applied to the feature slots of user input. We describe details of each
process below.

2.1. Feature slots and rules

We represent training and testing items as feature slots for the context in a sentence. A rule consists of a combination of a preposition and such feature slots. They are extracted from parsing sentences automatically. As an example:

(i) I decided not to take part in the event.

The feature slots and a rule extracted from preposition “in” in sentence (i) are as shown in Figure 2. The rule has a target preposition and feature slots that consist of two categories, Preceding and Following. The Preceding category means information about context preceding the target preposition. In this category, the Phrase type slot indicates a phrase to which the target prepositional phrase belongs. The Preposition slot is filled if another preposition exists before the target preposition, such as “up to.” The Following category contains head noun information following the target preposition.

A requirement when applying rules to new text is that the rule feature slots and target preposition’s context must agree. If the slot element is “-”, it counts as agreeing with any other element.

2.2. Extract rules for learning

It is impossible for the system to generate rules from the whole of a large corpus (say over a million words) because of the exponential growth of the recursive IL. Therefore, the system extracts moderate amounts of rules for IL from a rule database which is constructed by rules extracted from a large corpus directly in advance.

When a user sentence is input and its feature slots are agreed with the rule feature slots, the system extracts rules from the rule database in descending order of the number of the same slot elements (in Preceding Head, Verb, Preposition and Following Head slots) as user sentence feature slots. Ranked in this order, it is possible to extract rules in descending order of relevance to the target prepositional phrase context, which helps the system to generate effective rules.

2.3. Inductive Learning

In this paper, Inductive Learning is used as a method for discovering inherent regularities from actual examples [1]. The actual examples in our case are the rules extracted from a training large corpus directly. In the IL process, abstract rules are generated one after another by extracting common and different elements recursively from comparison of two rules’ feature slots. This process aims to generate rules which have only context elements dominating preposition selection.

Figure 3 shows an example of a new rule being generated by IL. In Figure 3, the two upper rules are extracted from sentence (i) and the sentence (ii) below. The new rule is generated by IL from the two upper rules.

(ii) I took part in the company sports meet.

The two upper rules have many common slot elements: other than Following Head. IL thus generates a new rule which has these common slot elements and abstract elements at the differing slots. In Figure 3, elements with “*” mean the slot is allowed to agree with any element, but not allowed to agree with an empty element.
If the only requirement for IL was that two rules must have common slot elements, too much abstracted rules would be generated. Thus, we add the requirements that the preposition elements of the rules must agree and that Preceding Head, Verb, Preposition or Following Head must also agree. IL then generates rules recursively until no rules meeting these requirements remain.

2.4. Calculation of scores for rules

After extracting and generating rules, the system calculates reliability scores for each rule using rules from the database. The rules for calculating scores have at least one the same slot element as user’s input in Preceding Head, Preposition and Following Head slots.

Now we define “NA (Number of times Applied)” and “NAC (Number of times Applied Correctly).” NA means the number of times a rule is applicable (context matches feature slots) in the rule database. NAC means the number of times the rule is applicable and the preposition used is the same as the one suggested by the rule. The score of the rules is defined by the following formula:

$$\text{score} = \frac{NAC}{NA}$$

We set a threshold parameter $\theta$ for the system. The system uses only rules with scores greater than or equal to $\theta$. All of the rules with the score is greater than or equal to $\theta$ are used for error correction, therefore, the system may suggest several prepositions.

3. Experiments

3.1. Training corpus and test data

We use the Reuters Corpus (about 256 million words) as a source of the rule database. The test data consists of 9 files from the Reuters Corpus which are not parts of the rule database. 201 prepositions are included in the test data. Since the test data does not include preposition errors, we evaluate whether the system suggests the same preposition as in the test data.

3.2. Experimental procedures

We compare system results to the test data for $\theta = 0$ to 1 with 0.2 increments in between, and evaluate its accuracy and coverage. The accuracy means the proportion of suggesting prepositions correctly to prepositions for which
4. Conclusions

In this paper, we proposed a system for correcting preposition errors automatically. The experiment results showed that the proposed system suggests correct prepositions with high accuracy (82.28%), which are competitive with other systems. And the system can control weight of precision or recall by threshold $\theta$.

However, our experiments are not enough in the following respects. First, the size of test data is too small. Tetreault et al. [8] used over 8,000 preposition contexts for test data, but we used only 200. Second, our experiments are not about error correction. We try to evaluate the system with larger test data including preposition errors in the near future.

### References


