

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

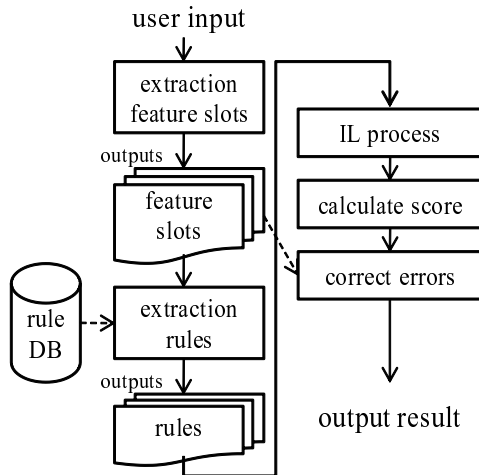


Figure 1. Process flow.

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

Preceding	Phrase type	<i>VP</i>
	Head	<i>part</i>
	Head POS	<i>NN</i>
	Modifier	-
	Verb	<i>take</i>
	Preposition	-
Following	Head	<i>event</i>
	Head POS	<i>NN</i>

⇒

Preposition
<i>in</i>

*) Slot element “-” means that no corresponding element is present.

Figure 2. An example of a rule.

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 created, 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.

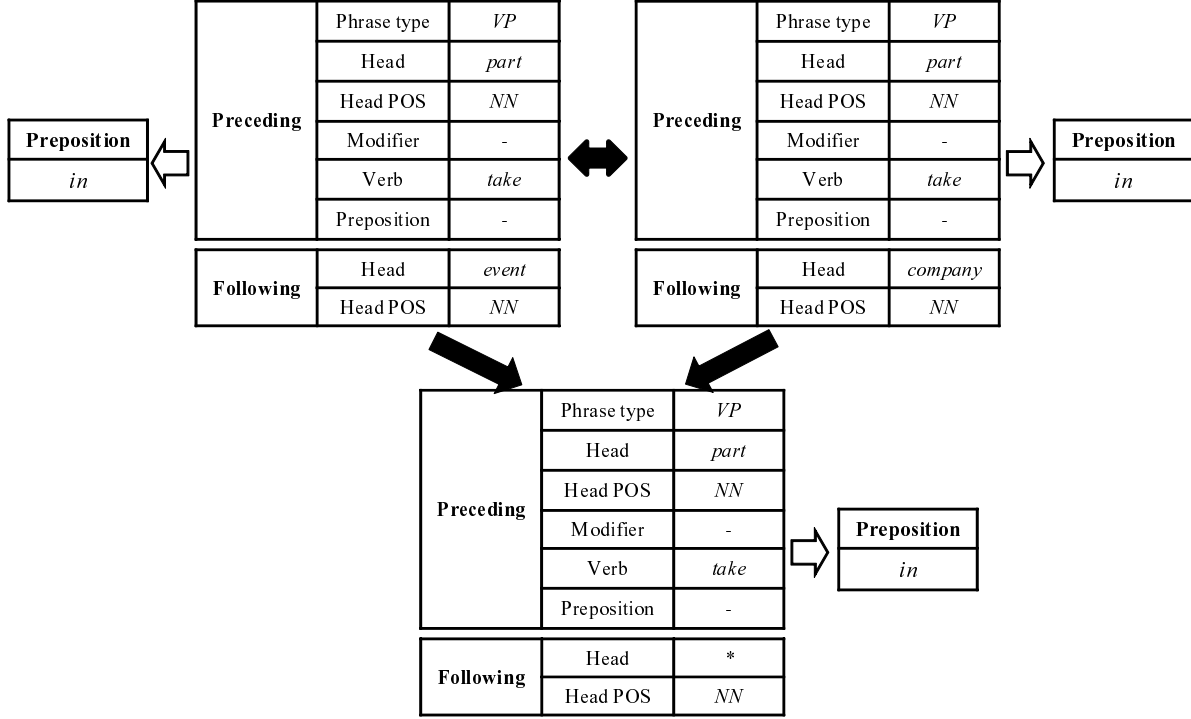


Figure 3. An example of when the IL process generates a new abstract rule.

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:

$$score = \frac{NAC}{NA} \quad (1)$$

We set a threshold parameter θ for the system. The system uses only rules with scores greater than or equal to θ . All of the rules with the score is greater than or equal to θ 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

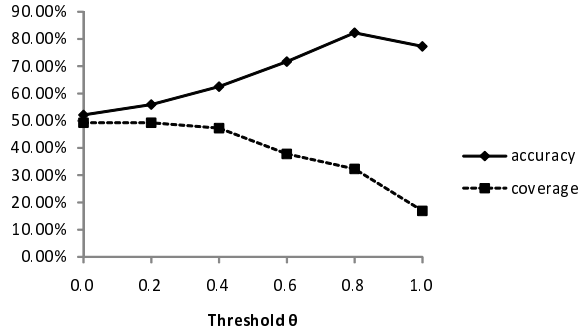


Figure 4. θ , accuracy and coverage.

at least one rule matches the context. The coverage is the proportion of suggesting prepositions correctly to all existing prepositions. A requirement of suggesting prepositions correctly is that the rule with the highest score suggests the correct preposition.

3.3. Results and discussion

Figure 4 shows the results of the experiments. In Figure 4, the threshold θ has much effect on the trade-off between the accuracy and the coverage. $\theta = 0.8$ gives the best accuracy 82.28% and then coverage is 31.34%. Therefore, in a preposition error correction task, there is a possibility that the system achieves 80% precision and 30% recall. Also, these results appear to have better recall than the results of Tetreault et al. [8] with 82.1% precision and 14.1% recall, although direct comparisons are difficult because of different training and test data.

In Figure 4, only when θ is incremented from 0.8 to 1.0, the accuracy degrades about 5%. We consider its reason is rules whose scores are calculated not enough because of small number of times applied. The system gives default score 1.0 (maximum) to a newly generated rule. If rules for calculating the score do not exist, the rule's reliability is not calculated with score 1.0. In $\theta = 1.0$, the system does not use rules whose reliability scores are calculated enough because of large number of times applied but even a little less than 1.0. To solve this problem, we try to use 0.0 as a default score and to prepare a more large corpus for a rule database.

There are still about 70% of the prepositions that could not be suggested correctly with $\theta = 0.8$. However, in $\theta = 0.0$, almost all prepositions in the test data are applied by at least one rule. Calculating scores with more rules, more precise reliabilities can be calculated. Therefore, rules with low scores may be useful by the reliability recalculation.

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 θ .

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

- [1] K. Araki and K. Tochinal. Effectiveness of natural language processing method using inductive learning. In *Proceedings of IASTED International Conference Artificial Intelligence and Soft Computing*, pages 295–300, Mexico, 2001.
- [2] R. D. Felice and S. G. Pulman. A classifier-based approach to preposition and determiner error correction in l2 english. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176, Manchester, UK, August 2008.
- [3] M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. B. Dolan, D. Belenko, and L. Vanderwende. Using contextual speller techniques and language modeling for esl error correction. In *Proceedings of IJCNLP*, Hyderabad, India, January 2008.
- [4] E. Izumi, K. Uchimoto, and H. Isahara. The overview of the sst speech corpus of japanese learner english and evaluation through the experiment on automatic detection of learners' errors. In *Proceedings of LREC*, pages 1435–1438, Lisbon, Portugal, May 2004.
- [5] E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara. Automatic error detection in the japanese learners' english spoken data. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 145–148, Sapporo, Japan, July 2003.
- [6] D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, pages 3–10. MIT Press, 2002.
- [7] H. Ototake and K. Araki. Correcting and detecting article errors method in english using inductive learning from conditions of word appearance. *IEICE Trans. Inf.&Syst.*, J90-D(6):1592–1601, June 2007.
- [8] J. R. Tetreault and M. Chodorow. The ups and downs of preposition error detection in esl writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872, Manchester, UK, August 2008.