4C1-IOS-4b-6

A machine learning-based approach to missing preposition detection

Shunsuke Ohashi*1 Tadayoshi Hara*2 Akiko Aizawa*3

*1 *3The University of Tokyo *2 *3 National Institute of Informatics

Estimated to exceed one billion, the number of those currently studying English as a foreign language is expected to continue growing. Various tools that draw on natural language processing have been developed to help students of English detect and correct writing errors. As one example, tools that correct spelling errors have achieved high accuracy and are now widely used throughout the world. These tools are used not just by those learning English, but in other areas - for natural language processing systems, for instance - to improve output from machine translation systems. Nevertheless, the tools for many other aspects, including grammar checking, remain relatively ineffective. This paper proposes a system for detecting missing prepositions based on syntactic information provided by an English language parser. The information lets us focus on locations that may lack a required preposition. This information can also be used as a machine learning feature to determine whether a location truly requires a preposition. By comparing the detection accuracy achieved against a simple baseline system, our study assessed the effectiveness of our system on the Konan-JIEM Learner Corpus, a typical English learner corpus. We found that our system achieved accuracy superior to the baseline system.

1. Introduction

Globalization has made English an international language whose importance continues to grow not just in academic settings, but in the world of business, where English now holds the status of a *lingua franca*.

The importance of English means there are many English learners all over the world. In many countries, including Japan, English is the first or primary foreign language taught as part of the compulsory curriculum. Clearly, English learners hope to progress to levels of proficiency at which they can use English effectively.

This has created high demand for tools and techniques that aid and support those learning or using English as a second language (ESL). One significant area in which one would expect natural language processing (NLP) to contribute is detecting and correcting grammar errors. Unfortunately, most such tools are intended for use by native speakers. Few have non-native speakers in mind. ESL learners make different kinds of errors than do native English speakers, and while prepositions constitute among the most difficult aspects of English for non-native speakers, the technology available for correcting prepositional errors in ESL text remains relatively unsophisticated.

Nevertheless, systems that correct prepositional errors exist. These systems focus mainly on detecting and correcting the misuse of prepositions, although the task of suggesting or inserting missing prepositions is equally important. Detecting the wrong choice of prepositions merely entails checking the prepositions already present in the text. Detecting the absence of a required preposition is more difficult, since so many word gaps can potentially accommodate a preposition. Due to this key difference, detecting missing prepositions is significantly more difficult than detecting incorrect preposition choices.

Contact: Shunsuke Ohashi, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8654, Japan, sohashi@nii.ac.jp

In this paper, we propose a method for detecting missing prepositions in text written by ESL students. Our method applies contextual information and trains a classifier that determines the presence or absence of a required preposition based on features extracted from contextual information. In the training phase of our method, we used raw text borrowed from Wikipedia*1 Our method did not require an annotated corpus.

We evaluated our proposal by implementing the idea and measuring the performance of our implementation and n-gram based baseline systems. In our experiments, our classifier demonstrated markedly better performance than a baseline system.

2. Related Work

2.1 ESL error correction model

In 2009, Gamon et al. [1] proposed a grammatical error correction method for text written by ESL students. This method involves identifying the potential insertion point preceding each noun phrase and extracts contextual information. The researchers trained two separate classifiers: one for presence/absence and another for choice. The presence/absence classifier determines whether a preposition should be present; if a preposition is required, the choice classifier determines what preposition is chosen. Each correction is examined by a language model, and corrections that reduce the language model score are filtered out. This is the strategy underlying much of the research on correcting errors in ESL writing. Their system achieved robust performance in correcting artificial errors.

2.2 Workshops for ESL error correction

Workshops on ESL error correction focus on correcting grammar errors. In these workshops, many systems that achieve high performance adopt the classification method, extracting contextual information from around the target

*1 http://en.wikipedia.org/

words to predict the appropriate correction based on a machine learned classifier, exploiting Web text resources, or using both approaches.

The Helping Our Own Exercise (HOO) [2] is a shared task whose goal is to assist computational linguistics researchers by applying tools based on computational linguistics. HOO 2012 focuses on correcting prepositions and determiners in non-native English writing.

The Error Detection and Correction Workshop (EDCW) is another workshop on ESL error correction. As part of this workshop, several teams tackled the challenge of preposition error correction. None of the methods adequately addressed errors involving missing prepositions.

3. Method

3.1 Overview

Figure 1 presents an overview of our method. Our method entails the following steps: (1) Spelling corrector; (2) parser; (3) classifier; (4) Language Model (LM) filter; and (5) LM detector. Each step is described in detail later.

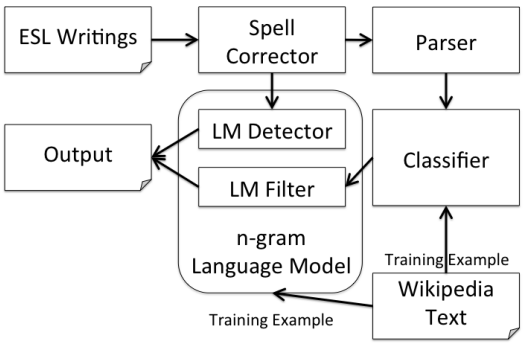

Figure 1: The overview of the system architecture

3.2 Spelling corrector

Before feature extraction, our method performs spelling corrections with GNU Aspell 0.60.6.1. Text written by ESL students contains many more misspellings than text written by native speakers. In addition to constituting actual textual errors, misspellings reduce the ability of the method to detect other errors, due to the resulting confusion of the classifier module.

First, the spelling correction module tokenizes the input text using the NLTK 2.0 [3] Python package for natural language processing (NLP). Second, GNU Aspell is applied for each token. If GNU Aspell detects a misspelling and supplies at least one suggested correction, the token is replaced with the suggestion by GNU Aspell.

3.3 Parser

Since we focused on noun phrases that are verb objects (in a V-N relation), the classifier module looked for missing prepositions before these noun phrases. The next section details how we used these relationships and detection examples. We used Enju [4], a syntactic parser for English, to identify this relationship.

3.4 Classifier

We treated the missing preposition detection problem as a binary classification problem. In each instance, the classifier determines the need for and the presence or absence of a preposition. We used libsvm 3.14 [5], a Support Vector Machine (SVM) implementation, as a classifier. The classifier training algorithm in our method was Soft Margin SVM.

To train the classifier module, we used raw sentences borrowed from Wikipedia. The idea here is that a particular noun phrase of interest would be a prepositional phrase if preceded by a preposition. We generated a negative (absence) training example from a noun phrase serving as the argument of a verb and a positive (presence) example from a prepositional phrase serving as the argument of a verb.

We extracted features from the contextual information to determine the absence or presence of a preposition.

We used the following features for the classifier:

Feature	*Example*
Words around the object word	have, developed, found, the, grammatical, error
Head of the noun phrase	error
POS tag of the head noun	NN
Head of the verb phrase	find
POS tag of the head verb	VBD

Table 1: Feature list and the example

Feature examples for the sentence "The system we have developed found the grammatical error." is also shown in Table 1. Words in a 3 word window around the front of the object word are taken as the *words around the object word*. The head words of these phrases are key, since the important grammatical behavior of a phrase depends on its head word.

3.5 Language model filter

Following detection, the phrases were filtered by an n-gram language model. We used SRI Language Modeling Toolkit (SRILM) 1.7.0 [6] to build a 3-gram language model from Wikipedia text. For each instance in which a missing preposition was detected, we attempted to insert each of the following 10 prepositions: on, in, at, for, of, about, from, to, by, with. Each potential detection was accepted only if the insertion increased the log probability of the sentence with a difference beyond a certain threshold (*filter strength*).

3.6 Language model detector

For each position at which the classifier made no decision regarding the absence or presence of a preposition, we used a language model to determine if a preposition insertion would increase the sentence score. If the insertion increased the score beyond a certain threshold (the *detection threshold*), the system concluded it had detected a missing preposition at that position. (The threshold here differs from the threshold used in the language model filter described above.)

4. Experiments

4.1 Data set

The Wikipedia text used for classifier and language model training was captured on December 10, 2012. The volume of the text was 7.38 GB. Using unaltered Wikipedia text, we trained the N-gram Language model at n = 3. (Unless otherwise noted, the classifier was trained on 400 MB of Wikipedia text.)

As the target text, we used Konan-JIEM Learner Corpus Third Edition [7], an ESL writings corpus with manually annotated grammatical errors. This corpus contains 233 essays written by 25 Japanese college students, with 3,401 sentences and 26,884 tokens in all.

In keeping with the conventions observed in past studies in this field, we corrected all grammatical errors other than prepositional errors to exclude the effects of any other types of error. However, spelling errors were left uncorrected.

At the preprocessing stage, we applied a syntactic parser to the evaluation data set, counting the missing prepositions and identifying the corresponding V-N relationships. Of these V-N relations, 75 accounted for 205 missing prepositions.

4.2 Experiment configuration

To explore the effects of the different modules described thus far, we compared the performance of several systems shown in Table 2. We refer to the LM detector as the system baseline because that module functions as a simple n-gram correction system.

System	*Classifier*	*LM Filter*	*LM Detector*
Baseline	No	No	Yes
Classifier	Yes	No	No
Classifier+Filter	Yes	Yes	No
Combined	Yes	No	Yes
Combined+Filter	Yes	Yes	Yes

Table 2: Baselines and their configurations

We performed the four different experiments, as indicated below:

- Classifier performance analysis

  We focused specifically on V-N relations and explored the output generated by the different detection methods to assess the impact of the classifier module.

- Overall evaluation

  To streamline the evaluation of our method, we combined the classifier and the LM detector, then compared the combined system to the LM detector, which functioned as the baseline.

- Effect of training data size

  To explore the effects of training data size, we trained our classifier using training data extracted from Wikipedia text corpus of various sizes.

- Effect of filtering weight

  We ran our method at several filter strengths to explore the effects of the language model filter and to identify the effects on precision and recall against the filter strength.

5. Results and Discussion

5.1 Classifier performance analysis

Table 3 shows the results on V-N relations in the Konan-JIEM Learner Corpus. Only word gap that is followed by a direct object word of verb phrase is checked, since this experiment is designated to evaluate performance of classifier module.

Comparing Classifier to Baseline, we found that the classifier module is more effective than LM detector in terms of recall. The language model filter was also deemed effective. The high recall of the classifier and the tendency of the language model filter to improve precision are complementary characteristics. This fact explains the significant improvements in the f-measure for Classifier+Filter compared to Baseline.

System	*Precision*	*Recall*	*F-measure*
Baseline	0.371	0.307	0.336
Classifier	0.305	0.627	0.410
Classifier+Filter	0.393	0.613	0.479

Table 3: Results of missing preposition detection for V-N relation

5.2 Overall evaluation

Table 4 shows the results for the entire Konan-JIEM Learner Corpus. Although the classifier module deals only with words that are direct objects, it is associated with better recall, as the classifier analysis experiment shows.

System	*Precision*	*Recall*	*F-measure*
Baseline	0.368	0.307	0.335
Combined	0.369	0.385	0.377
Combined+Filter	0.441	0.380	0.408

Table 4: Results of missing preposition detection for the entire text

5.3 Effect of corpus size

Figure 2 shows the relationship between corpus size for classifier and performance. For text of less than 100 MB, greater corpus size leads to greater precision, most likely due to the nature of the test data. While the number of noun phrases in the text data that are direct objects of the verb is 2,211, a mere 75 of these noun phrases actually require a preposition. Classifiers with insufficient training data are unable to handle this bias, and results for the output are no better than random guesses.

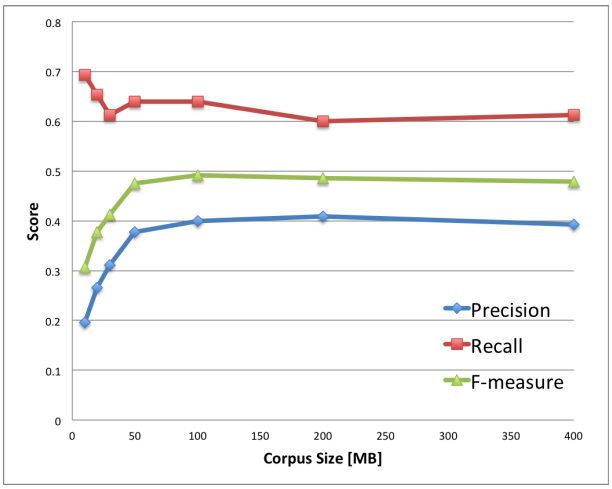

Figure 2: Effect of corpus size used for classifier training

5.4 Effects of language model filter

Figure 3 shows the relationship between the strength of the LM filter and performance. In essence, a stronger filter results in greater precision.The F-score has maximum with the filter what strength is 0.5.

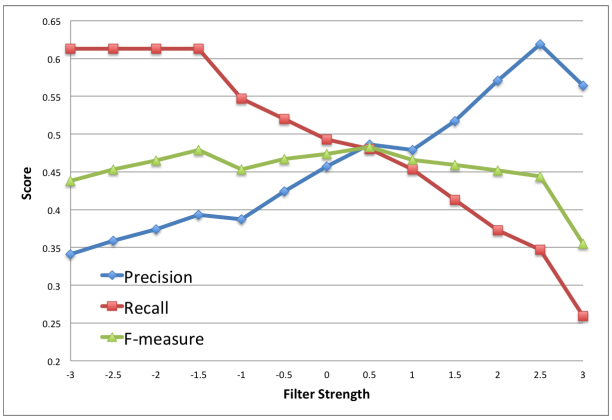

Figure 3: Effect of language model filter

5.5 Error analysis

This section addresses some of the errors resulting from our method. Our method assumed that all verbs have at most one object word, and we found that our method fails if the verb of the input sentence has multiple object words. Some output examples are shown below:

1. He talked *(wrong detect)/ε me *ε/about his life of Kyoto.
2. He took me *ε/to Kyoto University.

There is one unwanted detection and two missing detections in sentences 1 and 2. In sentence 1, the parser correctly recognizes the word "me" as the direct object of the word "talked" and identifies the relationship between "talked" and "his life of Kyoto," but our system ignored "his life of Kyoto," since we assumed there was not more than one object word. Missed detections in sentence 2 can be explained in the same way.

6. Conclusion

We proposed, implemented, and assessed a system for detecting missing prepositions, concluding that it performed comparatively better than previous attempts.

The following are areas in which future work may improve the method proposed. Our method produced relatively modest identification of missing prepositions. Training classifiers for noun phrases other than the direct object of a verb should improve performance. The model appears too simple to account for the complexity of determining whether a preposition is warranted or unwarranted. A more complex model should improve performance. Additionally, our method disregards information on the characteristics typical of errors made by ESL students. Using text written by ESL students as the classifier training corpus should improve performance.

References

[1] 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. *Urbana*, 51:61801, 2009.

[2] R. Dale and A. Kilgarriff. Helping our own: Text massaging for computational linguistics as a new shared task. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 263–267. ACL, 2010.

[3] S. Bird. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. ACL, 2006.

[4] Y. Miyao and J. Tsujii. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 83–90. ACL, 2005.

[5] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[6] A. Stolcke et al. SRILM-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904, 2002.

[7] R. Nagata, E. Whittaker, and V. Sheinman. Creating a manually error-tagged and shallow-parsed learner corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1210–1219, 2011.