The Companion Volume to the Proceedings of 41st Annual Meeting of the
Association for Computational Linguistics, July 2003, pp. 145-148.

Automatic error detection in the Japanese learners' English spoken data

Emi IZUMI[†‡] emi@crl.go.jp
Kiyotaka UCHIMOTO[†] uchimoto@crl.go.jp
Toyomi SAIGA[※] hoshi@karl.tis.co.jp
Thepchai Supnithi[*] thepchai@nectec.or.th
Hitoshi ISAHARA[†‡] isahara@crl.go.jp

[†] Computational Linguistics Group, Communications Research Laboratory,
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan
[‡] Graduate School of Science and Technology, Kobe University, 1-1 Rokkodai, Nada-ku, Kobe, Japan
[※]TIS Inc., 9-1 Toyotsu, Suita, Osaka, Japan
[*]National Electronics and Computer Technology Center,
112 Pahonyothin Road, Klong 1, Klong Luang, Pathumthani, 12120, Thailand

Abstract

This paper describes a method of detecting grammatical and lexical errors made by Japanese learners of English and other techniques that improve the accuracy of error detection with a limited amount of training data. In this paper, we demonstrate to what extent the proposed methods hold promise by conducting experiments using our learner corpus, which contains information on learners' errors.

1 Introduction

One of the most important things in keeping up with our current information-driven society is the acquisition of foreign languages, especially English for international communications. In developing a computer-assisted language teaching and learning environment, we have compiled a large-scale speech corpus of Japanese learner English, which provides a great deal of useful information on the construction of a model for the developmental stages of Japanese learners' speaking abilities.

In the support system for language learning, we have assumed that learners must be informed of what kind of errors they have made, and in which part of their utterances. To do this, we need to have a framework that will allow us to detect learners' errors automatically.

In this paper, we introduce a method of detecting learners' errors, and we examine to what extent this could be accomplished using our learner corpus data including error tags that are labeled with the learners' errors.

2 SST Corpus

The corpus data was based entirely on audio-recorded data extracted from an interview test, the "Standard Speaking Test (SST)". The SST is a face-to-face interview between an examiner and the test-taker. In most cases, the examiner is a native speaker of Japanese who is officially certified to be an SST examiner. All the interviews are audio-recorded, and judged by two or three raters based on an SST evaluation scheme (SST levels 1 to 9). We recorded 300 hours of data, totaling one million words, and transcribed this.

2.1 Error tags

We designed an original error tagset for learners' grammatical and lexical errors, which were relatively easy to categorize. Our error tags contained three pieces of information, i.e., the part of speech, the grammatical/lexical system and the corrected form. We prepared special tags for some errors that cannot be categorized into any word class, such as the misordering of words. Our error tagset currently consists of 45 tags. The following example is a sentence with an error tag.

```
*I lived in <at
crr="">the</at> New Jersey.
```

`at` indicates that it is an article error, and `crr=""` means that the corrected form does not need an article. By referring to information on the corrected form indicated in an error tag, the system can convert erroneous parts into corrected equivalents.

3 Error detection method

In this section, we would like to describe how we proceeded with error detection in the learner corpus.

3.1 Types of errors

We first divided errors into two groups depending on how their surface structures were different from those of the correct ones. The first was an “omission”-type error, where the necessary word was missing, and an error tag was inserted to interpolate it. The second was a “replacement”-type error, where the erroneous word was enclosed in an error tag to be replaced by the corrected version. We applied different methods to detecting these two kinds of errors.

3.2 Detection of omission-type errors

Omission-type errors were detected by estimating whether or not a necessary word string was missing in front of each word, including delimiters. We also estimated to which category the error belonged during this process. What we call “error categories” here means the 45 error categories that are defined in our error tagset. (e.g. article and tense errors) These are different from “error types” (omission or replacement). As we can see from Fig. 1, when more than one error category is given, we have two ways of choosing the best one. Method A allows us to estimate whether there is a missing word or not for each error category. This can be considered the same as deciding which of the two labels (E: “There is a missing word.” or C: “There is no missing word.”) should be inserted in front of each word. Here, there is an article missing in front of “telephone”, so this can be considered an omission-type error, which is categorized as an article error (“at” is a label that indicates that this is an article error.). In Method B, if *N* error categories come up, we need to choose the most appropriate error category “k” from among *N+1* categories, which means we have added one more category (*+1*) of “There is no missing word.” (labeled with “C”) to the *N* error categories. This can be considered the same as putting one of the *N+1* labels in front of each word. If there is more than one error tag inserted at the same location, they are combined to form a new error tag.

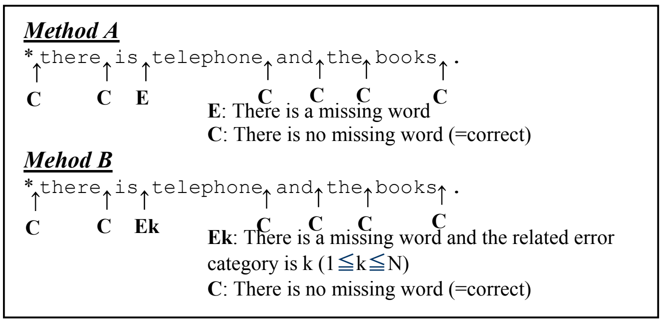

Method A
*there is telephone and the books .
C C E C C C C
E: There is a missing word
C: There is no missing word (=correct)
Mehod B
*there is telephone and the books .
C C Ek C C C C
Ek: There is a missing word and the related error category is k (1≦k≦N)
C: There is no missing word (=correct)

Figure 1. Detection of omission-type errors when there are more than one (N) error categories.

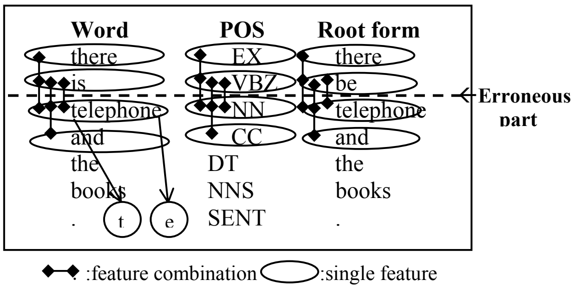
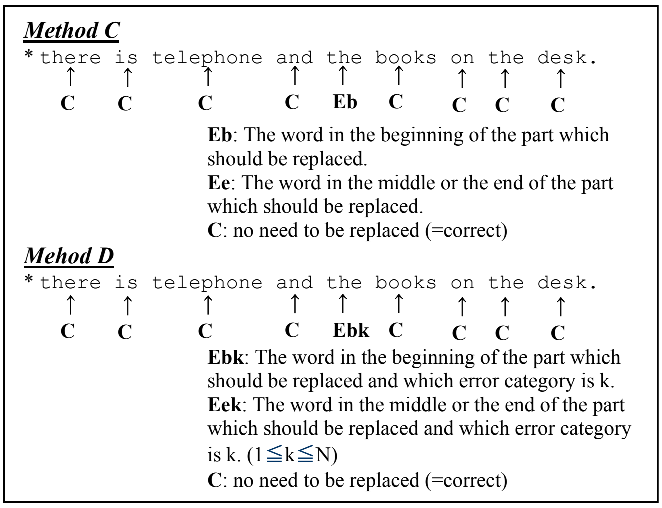


Figure 2. Features used for detecting omission-type errors

As we can see from Fig. 2, we referred to 23 pieces of information to estimate the error category: two preceding and following words, their word classes, their root forms, three combinations of these (one preceding word and one following word/two preceding words and one following word/one preceding word and two following words), and the first and last letter of the word immediately following. (In Fig. 2, “t” and “e” in “telephone”.) The word classes and root forms were acquired with “TreeTagger”. (Shmid 1994)

3.3 Detection of replacement-type errors

Replacement-type errors were detected by estimating whether or not each word should be deleted or replaced with another word string. The error category was also estimated during this process. As we did in detecting omission-type errors, if more than one error category was given, we use two methods of detection. Method C was used to estimate whether or not the word should be replaced with another word for each error category, and if it was to be replaced, the model estimated whether the word was located at the beginning, middle or end of the erroneous part. As we can see from Fig. 3, this can be considered the same as deciding which of the three labels (Eb: "The word is at the beginning of the erroneous part.", Ee: "The word is in the middle or end." or C: "The word is correct.") must be applied to each word. Method D was used if *N* error categories came up and we chose an appropriate one for the word from among *2N+1* categories. "*2N+1* categories" means that we divided *N* categories into two groups, i.e., where the word was at the beginning of the erroneous part and where the word was not at the beginning, and we added one more where the word neither needed to be deleted nor replaced. This can be considered the same as attaching one of the *2N+1* labels to each word. To do this, we applied Ramshaw's IOB scheme (Lance 1995). If there was more than one error tag attached to the same word, we only referred to the tag that covered the highest number of words.


Method C
* there is telephone and the books on the desk.
C C C C Eb C C C C
Mehod D
* there is telephone and the books on the desk.
C C C C Ebk C C C C

Eb: The word in the beginning of the part which should be replaced.
Ee: The word in the middle or the end of the part which should be replaced.
C: no need to be replaced (=correct)

Ebk: The word in the beginning of the part which should be replaced and which error category is k.
Eek: The word in the middle or the end of the part which should be replaced and which error category is k. (1≦k≦N)
C: no need to be replaced (=correct)

Figure 3. Detection of replacement-type errors when there are more than one (N) error categories.

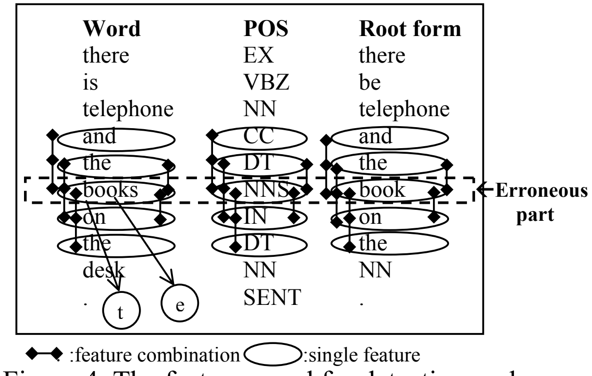


Figure 4. The features used for detecting replacement-type errors

As Fig. 4 reveals, 32 pieces of information are referenced to estimate an error category, i.e., the targeted word and the two preceding and following words, their word classes, their root forms, five combinations of these (the targeted word, the one preceding and one following/ the targeted word and the one preceding/ the targeted word and the one following/ the targeted word and the two preceding/ the targeted word and the two following), and the first and last letters of the word.

3.4 Use of machine learning model

The Maximum Entropy (ME) model (Jaynes 1957) is a general technique that is used to estimate the probability distributions of data. The over-riding principle in ME is that when nothing is known, the distribution should be as uniform as possible, i.e., maximum entropy. We calculated the distribution of probabilities p(a,b) with this method when Eq. 1 was satisfied and Eq. 2 was maximized. We then selected the category with maximum probability, as calculated from this distribution of probabilities, to be the correct category.

$$\sum_{a\in A,b\in B} p(a,b)g_j(a,b) = \sum_{a\in A,b\in B}\tilde{p}(a,b)g_j(a,b) \qquad (1)$$

$$\text{for } \forall f_j \ (1 \le j \le k)$$

$$H(p) = -\sum_{a\in A,b\in B} p(a,b)\log(p(a,b)) \qquad (2)$$

We assumed that the constraint of feature sets f_i *(i≦j≦k)* was defined by Eq. 1. This is where *A* is a set of categories and *B* is a set of contexts, and $g_j(a,b)$ is a binary function that returns value 1 when feature f_j exists in context *b* and the category is *a*. Otherwise, $g_j(a,b)$ returns value 0. $\tilde{p}(a,b)$ is the occurrence rate of the pair *(a,b)* in the training data.

4 Experiment

4.1 Targeted error categories

We selected 13 error categories for detection.

Table 1. Error categories to be detected

Noun	Number error, Lexical error
Verb	Erroneous subject-verb agreement, Tense error, Compliment error
Adjective	Lexical error
Adverb	Lexical error
Preposition	Lexical error on normal and dependent preposition
Article	Lexical error
Pronoun	Lexical error
Others	Collocation error

4.2 Experiment based on tagged data

We obtained data from 56 learners' with error tags. We used 50 files (5599 sentences) as the training data, and 6 files (617 sentences) as the test data.

We tried to detect each error category using the methods discussed in Sections 3.2 and 3.3. There were some error categories that could not be detected because of the lack of training data, but we have obtained the following results for article errors which occurred most frequently.

Article errors		
Omission-type errors	Recall rate	8/71 * 100 = 32.39(%)
	Precision rate	8/11 * 100 = 52.27(%)
Replacement-type errors	Recall rate	0/43 * 100 = 9.30(%)
	Precision rate	0/ 1 * 100 = 22.22(%)

Results for 13 errors were as follows.

All errors		
Omission-type errors	Recall rate	21/ 93 * 100 = 22.58(%)
	Precision rate	21/ 38 * 100 = 55.26(%)
Replacement-type errors	Recall rate	5/224 * 100 = 2.23(%)
	Precision rate	5/ 56 * 100 = 8.93(%)

We assumed that the results were inadequate because we did not have sufficient training data. To overcome this, we added the correct sentences to see how this would affect the results.

4.3 Addition of corrected sentences

As discussed in Section 2.1, our error tags provided a corrected form for each error. If the erroneous parts were replaced with the corrected forms indicated in the error tags one-by-one, ill-formed sentences could be converted into corrected equivalents. We did this with the 50 items of training data to extract the correct sentences and then added them to the training data. We also added the interviewers' utterances in the entire corpus data (totaling 1202 files, excluding 6 that were used as the test data) to the training data as correct sentences. We added a total of 104925 correct new sentences. The results we obtained by detecting article errors with the new data were as follows.

Article errors		
Omission-type errors	Recall rate	8/71 * 100 = 11.27(%)
	Precision rate	8/11 * 100 = 72.73(%)
Replacement-type errors	Recall rate	0/43 * 100 = 0.00(%)
	Precision rate	0/ 1 * 100 = 0.00(%)

We found that although the recall rate decreased, the precision rate went up through adding correct sentences to the training data.

We then determined how we could improve the results by adding the artificially made errors to the training data.

4.4 Addition of sentences with artificially made errors

We did this only for article errors. We first examined what kind of errors had been made with articles and found that "a", "an", "the" and the absence of articles were often confused. We made up pseudo-errors just by replacing the correctly used articles with one of the others. The results of detecting article errors using the new training data, including the new corrected sentences described in Section 4.2, and 7558 sentences that contained artificially made errors were as follows.

Article errors		
Omission-type errors	Recall rate	24/71 * 100 = 33.80(%)
	Precision rate	24/30 * 100 = 80.00(%)
Replacement-type errors	Recall rate	2/43 * 100 = 4.65(%)
	Precision rate	2/ 9 * 100 = 22.22(%)

We obtained a better recall and precision rate for omission-type errors.

There were no improvements for replacement-type errors. Since some more detailed context might be necessary to decide whether "a" or "the" must be used, the features we used here might be insufficient.

5 Conclusion

In this paper, we explained how errors in learners' spoken data could be detected and in the experiment, using the corpus as it was, the recall rate was about 30% and the precision rate was about 50%. By adding corrected sentences and artificially made errors, the precision rate rose to 80% while the recall rate remained the same.

References

Helmut Schmid Probabilistic part-of-Speech tagging using decision trees. *In Proceedings of International Conference on New Methods in Language Processing*. pp. 44-49, 1994.

Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. *In Proceedings of the Third ACL Workshop on Very Large Corpora*, pp. 82-94, 1995.

Jaynes, E. T. "Information Theory and Statistical Mechanics" Physical Review, 106, pp. 620-630, 1957.