The overview of the SST speech corpus of Japanese learner English and evaluation through the experiment on automatic detection of learners' errors

Emi Izumi[†‡], **Kiyotaka Uchimoto**[†], **and Hitoshi Isahara**[†‡]

[†]Computational Linguistics Group, National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan
[‡]Graduate School of Science and Technology, Kobe University
1-1 Rokkodai, Nada-ku, Kobe, Japan
{emi, uchimoto, isahara}@crl.go.jp

Abstract

This paper introduces an overview of the speech corpus of Japanese learner English compiled by National Institute of Information and Communications Technology by showing its data collection procedure and annotation schemes including error tagging. We have collected 1,200 interviews for three years. One of the most unique features of this corpus is that it contains rich information on learners' errors. We have performed error tagging for learners' grammatical and lexical errors with originally-designed error tagset. We also evaluated the corpus through the experiment on automatic detection of learners' errors by using error tag information in the corpus. We did this by using a machine learning model, Maximum Entropy (ME) model. Since we had obtained the limited amount of error-tagged data, we needed to make some efforts to enlarge training data. We added the correct sentences and artificially-made errors to the training data, and found that it improved accuracy. We are planning to make this corpus publicly available in the spring of 2004, so that teachers and researchers in many fields can use the data for their own interests, such as second language acquisition research, syllabus and material design, or the development of computerized pedagogical tools, by combining it with NLP technology.

1 Introduction

As corpus-based research has been flourished, the various kinds of corpora in various languages have been compiled in the world. A learner corpus is one of the new types. Although learner corpus research is becoming increasingly popular, most existing learner corpora focus on learners' written language. We have compiled one-million word corpus of Japanese learner English, focusing on the speaking skill which is the most difficult for Japanese learners to acquire. The corpus is entirely based upon the audio-recordings of an English oral proficiency interview test called the *Standard Speaking Test* (*SST*). We have collected 1,200 interviews for three years. One of the most unique features of this corpus is that it contains rich information on learners' errors. We have performed error tagging for learners' grammatical and lexical errors with the originally-designed error tagset. Error analysis, based on the error-tagged texts, can be a great help to develop an error diagnostic system, and this will enable the construction of a computer-assisted language learning (CALL) system that can accept learners' poorly-formed texts and provide them with feedback.

In this paper, firstly, we are going to give an overview of the *SST speech corpus of Japanese learner English*, by introducing its data acquisition procedures and annotation schemes. We will subsequently describe what we found through the experiments about to what extent this corpus can be exploited for automatic detection of learners' errors with a machine learning technique using error tag information.

2 The *SST speech corpus of Japanese learner English*

In this section, we will give an overview of the *SST speech corpus of Japanese learner English*, mainly by explaining the nature of the *SST* interview technique and the method by which learner data has been collected, transcribed, and annotated including error tagging. We will also mention the native speakers' data which were collected in order to observe learners' language from a broad perspective.

2.1 The *SST*

Firstly, we will describe some details of the *SST* which is a face-to-face interview test that measures the English speaking ability of Japanese learners. This 15-minute interview test comprises five parts, commencing with an informal chat on general topics such as the interviewees' job, hobbies, family, and so on. During the second to fourth stages of the interview, the interviewee is asked to perform three task-based activities, namely, picture description, role-playing, and story telling. The interview ends with another informal chat. All interviews are tape-recorded and rated at one of the nine proficiency levels (*SST* level one to nine) by two or three assessors based on the *SST*'s original evaluation scheme.

2.2 Recording ~ Transcribing

Each interview was recorded in a quiet room by means of DAT (Digital-Audio Tape) as the medium. There are some general rules for transcribing. For instance, even though a word may be mispronounced, it is transcribed with the correct spelling, provide that the transcribers are able to understand the word that was produced. If acronyms are pronounced as sequences of letters, they must be transcribed as a series of upper case letters, which are separated by spaces. Roman or Arabic numerals must not be used; all numbers must be transliterated as words. The transcribers are allowed to insert phrase and sentence boundaries with commas and periods, based on their own discretion. Some information on non-verbal behaviors or concurrent events such as relevant noises is also inserted.

2.3 Tagging

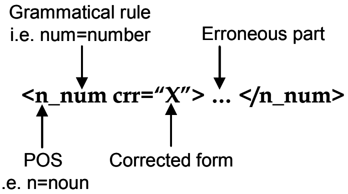
There are two kinds of tags used in this corpus: basic tags for discourse phenomena such as filled pauses or repetitions, and error tags for the analysis of the learners' errors. The tags are based on XML syntax.

2.3.1 Basic tagging

There are more than 30 basic tags for identifying discourse phenomena in the utterances. These are divided into four groups: tags for representing the structure of the interview, tags for the interviewee's profile, tags for speaker turns, and tags for representing utterance phenomena such as fillers, repetitions, self-corrections, overlapping, and so on.

2.3.2 Error tagging

Analyzing errors produced by learners is an efficient way of finding out the learners' stages of development and for deciding the most appropriate teaching method for them. We are aware that it is quite difficult to design a consistent and generic error tagset as the learners' errors extend across various linguistic areas. We need to have a robust error typology to accomplish this. We designed the original error tagset only for learners' grammatical and lexical errors, which are relatively easy to categorize, compared with other error types such as discourse errors or errors related to more communicative aspects of learners' language. The error tagset consists of 45 tags. As shown in Figure1, an error tag contains three pieces of information: part of speech, a grammatical and lexical rule, and a corrected form.



ex) *I belong to two baseball <n_num crr="teams">team</n_num>.

Figure1: Structure of an error tag and an example of an error-tagged sentence

By referring to the corrected form indicated in an error tag, it is possible to obtain a corrected sentence just by converting erroneous parts into corrected equivalents.

Since manual error tagging is very time-consuming task, we have obtained only 167 error-tagged transcripts so far.

2.4 Subcorpus

As stated in 2.3, we dealt only with the formal aspects of learners' language such as grammatical and lexical errors. In order to examine what we are unable to examine solely by error-tagged data, a native English speakers' corpus has been compiled. It has been made by collecting the speech data of native speakers' conducting a similar type of interview to that of the *SST*. It is considered to be quite useful for comparing the utterances of native speakers and Japanese learners.

3 Detection of learners' errors

The *SST speech corpus of Japanese learner English* will be able to be exploited in various research areas. One of the most practical applications might be the development of a computer-assisted language learning (CALL) system by applying NLP technology. In the support system for language learning, we have assumed that learners should be told 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 section, we are going to demonstrate several experiments on automatic error detection with machine learning technique. We will examine to what extent this could be accomplished using error tag information in our learner corpus, by describing the method for detecting learners' grammatical and lexical errors with a machine learning model, the Maximum Entropy (ME) model. Since we had obtained the limited amount of error-tagged data, we needed to make some efforts to enlarge training data. At the end of this section, we will also introduce our new system, "*Eden* (*Error Detection System for English*)" which has been developed based on these experiments.

3.1 Method

3.1.1 Types of errors

We first categorized learners' errors into three types depending on how their surface structures differ from those of the correct sentences. The first of these is an "omission-type" error, in which a necessary word is missing. The second is a "replacement-type" error, in which an erroneous word is used. The third is an "insertion-type" error, in which an extra word is used. The detection method of each type of error can be divided into two, depending on how error tags are labeled. One is for the detection of omission-type errors, where error tags are inserted to interpolate the missing word. The other is for replacement-type and insertion-type errors, where an erroneous word is enclosed in an error tag to be replaced by the correct word (replacement-type errors) or a zero element (insertion-type errors).

3.1.2 Detection of omission-type errors

Omission-type errors are detected by determining whether or not a necessary word or expression is missing in front of each word, including delimiters (Figure2, Method A). During this process, we also determine the category the error belonged to. The expression "error categories" here means the 45 error categories that have been defined in our error tagset (e.g. article errors, tense errors, and so on). It must be noted that "error categories" are different from "types of errors" mentioned in 3.1.1. If more than one error category is given, we need to choose the most appropriate error category "*k*" from among *N+1* categories, which means we have added one more category (*+1*), namely "There is no missing word." (labeled with "*C*") to the *N* error categories (Figure2, Method B).

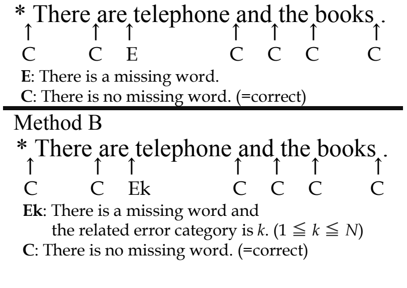
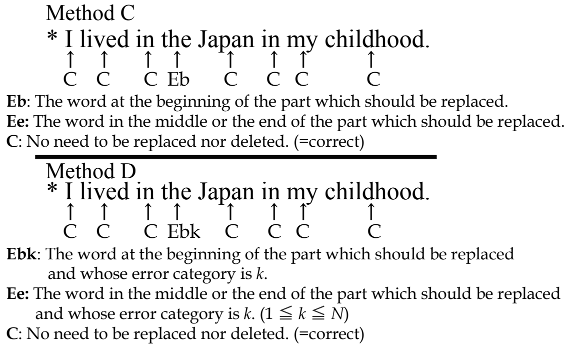

Method A
* There are telephone and the books .
C C E C C C C
E: There is a missing word.
C: There is no missing word. (=correct)
Method B
* There are telephone and the books .
C C Ek C C C C
Ek: There is a missing word and the related error category is k. (1 ≦ k ≦ N)
C: There is no missing word. (=correct)

Figure2: Detection of omission-type errors

To perform the estimation, we refer to 23 pieces of information as described. These are the 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 letters of the word immediately following the putative omission point (e.g. in Figure2, "*t*" and "*e*" in "*telephone*"). The word classes and root forms are obtained using "TreeTagger" (Schmid, 1994).

3.1.3 Detection of replacement-/insertion-type errors

Replacement-type and insertion-type errors are detected by estimating whether or not each word should be deleted or replaced with another word string. The error category is also determined during this process. If more than one error category is determined, we use two methods of detection as shown in Figure3. In Method C, if the word is to be replaced, the model estimates whether the word is located at the beginning, middle, or end of the erroneous part. Method D is used if *N* error categories arise. We choose an error category for the word from among *2N+1* categories. "*2N+1* categories" means that we divide *N* categories into two groups, i.e., firstly when the word is at the beginning of the erroneous part and secondly when the word is not at the beginning. We add one more (*+1*) when the word neither needs to be deleted nor replaced. To do this, we applied Ramshaw's IOB scheme (Ramshaw 1995).


Method C
* I lived in the Japan in my childhood.
C C C Eb C C C C
Method D
* I lived in the Japan in my childhood.
C C C Ebk C C C C

Eb: The word at 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 nor deleted. (=correct)

Ebk: The word at the beginning of the part which should be replaced and whose error category is *k*.
Ee: The word in the middle or the end of the part which should be replaced and whose error category is *k*. (1 ≦ *k* ≦ *N*)
C: No need to be replaced nor deleted. (=correct)

Figure3: Detection of replacement-/insertion-type errors

To estimate an error category, we refer to 32 pieces of information. These are the targeted word and the two preceding and two 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 targeted word.

3.1.4 Use of machine learning technique

The Maximum Entropy (ME) (Jaynes 1957, 1979) model is one of the general techniques for estimating probability distributions of data. The over-riding principle in ME is that when nothing is known, the distribution should be as uniform as possible, that is, have maximum entropy. As shown in Figure4, we calculate the distribution of probabilities *p(a,b)* when Eq. (1) is satisfied and Eq. (2) is maximized. The category with the maximum probability, as calculated from this distribution of probabilities, is selected to be the correct category.

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

$$for\ \forall f_j\ (1 \le j \le k)$$

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

Figure4: The Maximum Entropy Model

We assume that a constraint of feature sets f_i *(i≦j≦k)* is defined by Eq. (1). *A* is a set of categories and *B* is a set of contexts, $g_j(a,b)$ 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 the value 0. $\tilde{p}$ *(a,b)* is the occurrence rate of the pair *(a,b)* in the training data.

3.2 Experiment

3.2.1 Targeted error categories

As shown in Table1, we selected 13 error categories for detection. We assume that these errors are more frequent than other errors, and can be identified relatively easily from the context.

Noun	Number error, Lexical error
Verb	Erroneous subject-verb agreement, Tense error, Compliment error, Lexical error
Adjective	Lexical error
Adverb	Lexical error
Preposition	Lexical error (normal/dependent)
Article	Lexical error
Pronoun	Lexical error
Others	Collocational error

Table1: Error categories to be detected

3.2.2 Experiment1: Based on tagged data

We obtained 167 error-tagged transcripts from the SST Corpus. We used 151 files (16837 sentences) as training data, and 16 files (1915 sentences) as test data.

We tried to detect each error category using the method described in 3.1. Since there were some error categories that could not be detected due to the lack of training data, the overall rate was inadequate (Figure5). The best results were obtained for article errors, which were the most frequently occurring errors, as shown in Figure6.

All errors		
Omission-type	Recall	96/277 * 100= 34.66 %
	Precision	96/169 * 100= 56.88 %
Replacement-/Insertion-type	Recall	37/647 * 100= 5.72 %
	Precision	37/183 * 100= 20.22 %

Figure5: Recall/Precision for the detection of all errors

Article errors		
Omission-type	Recall	86/172 * 100= 50.00 %
	Precision	86/143 * 100= 60.14 %
Replacement-/Insertion-type	Recall	13/88 * 100= 14.77 %
	Precision	13/44 * 100= 29.55 %

Figure6: Recall/Precision for the detection of article errors

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

3.2.3 Experiment2: Addition of correct sentences

We added the correct sentences of the following two types. The first type is the native speaker subcorpus which was mentioned in 2.4. The second type is the corrected sentences extracted from the error-tagged data. Since our error tags provide a corrected form for each error, if the erroneous parts are replaced with the corrected forms indicated in the error tags individually, poorly-formed sentences can be converted into corrected equivalents. We extracted the corrected sentences from 151 error-tagged files. We added a total of approximately 30,000 new correct sentences.

By doing this, although the rates of recall in the detection of omission-type errors in all error categories decreased by 12%, the precision went up to 75.61%. The result remained steady for the detection of replacement and insertion-type errors (Figure7).

All errors		
Omission-type	Recall	62/277 * 100= 22.38 %
	Precision	62/82 * 100= 75.61 %
Replacement-/Insertion-type	Recall	37/647 * 100= 5.72 %
	Precision	37/183 * 100= 20.22 %

Figure7: Recall/Precision for the detection of all errors

For article errors, the recall of detecting omission-type errors decreased by 19%, but the precision went up by 16%. In the detection of replacement and insertion-type errors, the precision increased sharply to 60% (Figure8).

Article errors		
Omission-type	Recall	54/172 * 100= 31.40 %
	Precision	54/71 * 100= 76.06 %
Replacement-/Insertion-type	Recall	6/88 * 100= 6.82 %
	Precision	6/10 * 100= 60.00 %

Figure8: Recall/Precision for the detection of article errors

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

3.2.4 Experiment3: Addition of artificially-made errors

Article errors were automatically added by using simple manually-constructed rules. These rules were derived by investigating the characteristics of learners' errors found in our corpus. We first examined what kind of article errors had been made and found that there was often confusion between "*a*", "*an*", "*the*" and the absence of an article. We made up pseudo-errors by replacing the correctly used articles with one of the alternatives. The results using the new training data, including the new corrected sentences described in 3.2.3, and (7578) sentences that contained artificially-made errors, are shown in Figures9 and 10.

All errors		
Omission-type	Recall	89/277 * 100= 32.13 %
	Precision	89/122 * 100= 72.95 %
Replacement-/Insertion-type	Recall	46/647 * 100= 7.11 %
	Precision	46/183 * 100= 25.14 %

Figure9: Recall/Precision for the detection of all errors

Article errors		
Omission-type	Recall	89/172 * 100= 51.74 %
	Precision	89/116 * 100= 76.72 %
Replacement-/Insertion-type	Recall	19/88 * 100= 21.59 %
	Precision	19/25 * 100= 76.00 %

Figure10: Recall/Precision for the detection of article errors

We obtained a better recall and precision rate for all types of errors except for the recall rate in the detection of omission-type errors in all error categories. We found that adding the correct sentences or adding artificially-made errors to the training data improves accuracy. However, to improve accuracy for the detection of replacement and insertion-type errors, we need to obtain more error-tagged sentences and examine global context more thoroughly.

3.2.5 Summary of results

By using the corpus, in its original form, our experiment showed the recall of article errors to be 50% and the precision to be approximately 60%. By adding corrected sentences and artificially-made errors, recall and precision improved to 51% and 76%, respectively.

Minnen et al. (2000) proposed a method for determining whether or not an article should be used for a noun phrase and which article is appropriate by using memory-based learning. Newspaper articles that only contained a few errors were used for this purpose. Conversely, our learner data contains a number of different kinds of errors, and, of course, the errors can occur not only in noun phrases. Therefore, our method has been designed to detect errors of all kinds of words. We will examine to what extent our method can be improved by incorporating the new features used in Minnen et al.'s framework into our method.

4 Conclusion

In this paper, we have presented an overview of the *SST speech corpus of Japanese learner English*, by explaining data collection procedures such as transcribing and tagging, including error tagging. We have also illustrated how this corpus can be utilized by way of a framework for the automatic detection of learners' errors.

We are planning to make this corpus publicly available in the spring of 2004, so that teachers and researchers in many fields can use the data for their own interests, such as second language acquisition research, syllabus and material design, or the development of computerized pedagogical tools, by combining it with NLP technology.

References

Jaynes, E.T. (1957) Information theory and statistical mechanics. *Physical Review* 106 (pp. 620-630).

Jaynes E.T. (1979) Where do we stand on maximum entropy? In Levine, R.D. and M. Tribus (Eds.). *The maximum entropy formalism*. (pp. 15) M.I.T Press.

Minnen, G., Bond, F., and Copestake, A. (2000) Memory-based learning for article generation. *In Proceedings of CoNLL-2000 and LLL-2000*. (pp. 43-48).

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

Schmid, H. (1994) Probabilistic part-of-speech tagging using decision trees. *In Proceedings of International Conference on New Methods in Language Processing*. (pp. 45-49).