SST speech corpus of Japanese learners’ English and automatic detection of learners’ errors

Emi Izumi[a,b]*, Kiyotaka Uchimoto*[a] *and Hitoshi Isahara*[a,b]
[a] *Communications Research Laboratory*
[b] *Kobe University*

1 Introduction

To keep up with information-driven society, one of the most important tasks is to acquire foreign language skills, especially English, for international communication. Of the four main skills for language acquisition (reading, writing, listening, and speaking), we are focusing on speaking as it is the most difficult skill for Japanese learners to acquire. CLE (Corpus of Learner English) research is becoming increasingly popular in Japan, but most existing learner corpora focus on learners’ written language. In order to construct a model of the developmental stages of Japanese learners’ speaking ability, we decided to compile a large-scale speech corpus called “The SST Corpus”. Our corpus is entirely based upon the audio-recordings of an English oral proficiency interview test called the Standard Speaking Test (SST).

In this paper, firstly, we are going to give an overview of this new learner corpus by introducing the activities of the project to date, such as its data acquisition procedures and annotation schemes. We will subsequently describe the extent to which this corpus can be exploited for automatic detection of learners’ errors with a machine learning technique which is a probabilistic framework for classifications. At the end of this paper, we will also consider how and in what kind of research area this corpus can be utilized.

2 Overview of the SST Corpus

In this section, we will give an overview of the SST Corpus, mainly by explaining the nature of the SST interview technique and the method by which learner data has been collected, transcribed, and annotated. Two subcorpora have been compiled 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 test has been formulated for Japanese learners based on the Oral Proficiency Interview (OPI) that was originally developed mainly by the American Council for the Teaching of Foreign Language (ACTFL). The 15-minute interview test comprises five parts, commencing with an informal chat on general topics, such as the interviewee's 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. Each stage consists of two sections: a task and a follow-up. After the task, the interviewer asks some questions associated with the task in the follow-up section. The interview ends with another informal chat. All interviews are tape-recorded and judged by two or three assessors based on the SST evaluation scheme (SST levels 1 to 9).

We consider the SST to be a very useful spoken resource, since most existing learner corpora comprise only written language. Another benefit of choosing the SST as the main resource of the corpus is that each data file has specific information on the examinee's oral proficiency level, as assessed by professional examiners. There are some developmental learners' corpora available, but in most cases, these determine the learners' proficiency level based on external factors, such as their level of schooling. A comparison between sub-corpora based on schooling level may not be reliable because it ignores other factors in the students' background, such as the way in which they have been learning English and from what type of teachers. The SST data, on the other hand, contains more reliable information on learners' proficiency levels, which will assist in making comparative research based on proficiency subsections of the corpus more valid (Tono 2001).

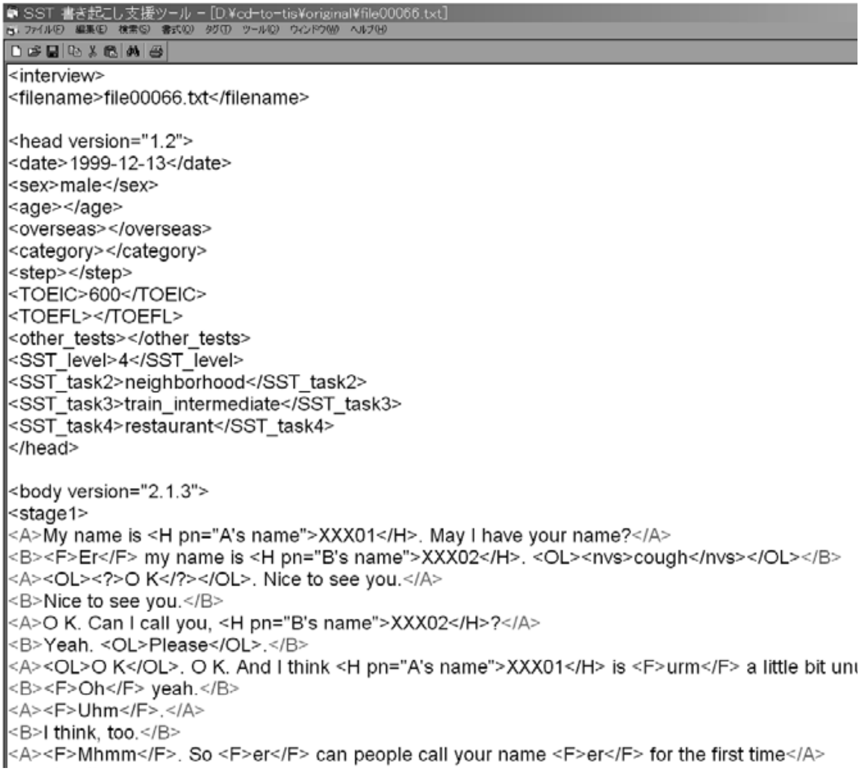
2.2 Recordings

Each interview was recorded in a quiet room by means of DAT (Digital-Audio Tape) as the medium. The use of a headset microphone would ideally be preferable for data quality, but we decided to use a table-top microphone as we considered that speakers might be uncomfortable with a headset microphone.

2.3 Transcription

There are some general rules for transcribing the speech data. For instance, even though a word may be mispronounced, it is transcribed with the correct spelling, provided 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.



```
<interview>
<filename>file00066.txt</filename>

<head version="1.2">
<date>1999-12-13</date>
<sex>male</sex>
<age></age>
<overseas></overseas>
<category></category>
<step></step>
<TOEIC>600</TOEIC>
<TOEFL></TOEFL>
<other_tests></other_tests>
<SST_level>4</SST_level>
<SST_task2>neighborhood</SST_task2>
<SST_task3>train_intermediate</SST_task3>
<SST_task4>restaurant</SST_task4>
</head>

<body version="2.1.3">
<stage1>
<A>My name is <H pn="A's name">XXX01</H>. May I have your name?</A>
<B><F>Er</F> my name is <H pn="B's name">XXX02</H>. <OL><nvs>cough</nvs></OL></B>
<A><OL><?>O K</?></OL>. Nice to see you.</A>
<B>Nice to see you.</B>
<A>O K. Can I call you, <H pn="B's name">XXX02</H>?</A>
<B>Yeah. <OL>Please</OL>.</B>
<A><OL>O K</OL>. O K. And I think <H pn="A's name">XXX01</H> is <F>urm</F> a little bit un
<B><F>Oh</F> yeah.</B>
<A><F>Uhm</F>.</A>
<B>I think, too.</B>
<A><F>Mhmm</F>. So <F>er</F> can people call your name <F>er</F> for the first time</A>
```

Figure 1: Transcription of a learner's speech data

2.4 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. One advantage of using XML is that it can clearly identify the structure of the text and it is also quite beneficial when corpus data is utilized for web-based pedagogical tools or database as a hypertext.

2.4.1 Discourse tagging

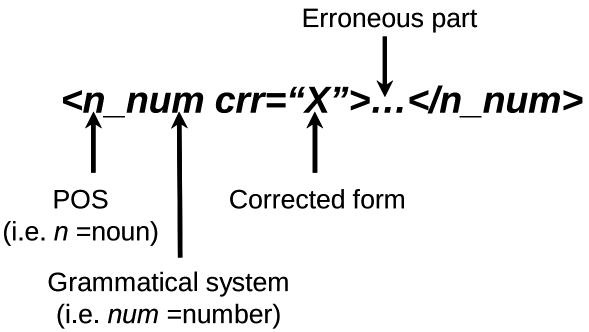
There are more than 30 basic tags for identifying discourse phenomena in the learners' utterances. These are divided into four categories: tags for representing the structure of the entire transcription file, tags for the interviewee's profile (which is attached as a header of the file), tags for speaker turns, and tags for representing utterance phenomena, such as fillers and repetitions. Some of the basic tags are shown in Table 1:

Table 1: The basic tags

Tag	**Function**
<F></F>	Filled pause
<R></R>	Repetition
<SC></SC>	Self-correction
<CO></CO>	Incomplete utterance
<?></?>	Unclear utterance but the words can be guessed from the context
<??></??>	Totally unclear utterance
<JP></JP>	Use of Japanese words
<.></.>	Short pause (2–3 sec.)
<..></..>	Long pause (more than 3 sec.)
<OL></OL>	Overlapping
<nvs></nvs>	Non-verbal sound
<ctxt></ctxt>	Events taking place simultaneously with the speech
<laughter> </laughter>	Utterance with a laugh

2.4.2 Error tagging

It has been said that 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. In this project, we decided to analyze errors mainly by error tagging, in order to construct a model of Japanese learners’ English across different proficiency levels. We are aware that it is quite difficult to design a consistent error tagset as the learners’ errors extend across various linguistic areas, including grammar, lexis and phonetics, and so on. In order to do this, it is necessary to have a robust error typology.



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

Figure 2: Structure of an error tag and an example of an error tagged sentence

We designed our 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. As shown in Figure 2, our error tags contain three pieces of information: part of speech, a grammatical/lexical rule, and a corrected form. For errors that cannot be categorized as belonging to any word class, such as the misordering of words, we prepare special tags. Our error tagset currently consists of 45 tags (Table 2).

Table 2: List of error tags

Tag	Error category
	NOUN
<n_inf>…</n_inf>	Noun inflection
<n_num>…</n_num>	Number of noun
<n_cs>…</n_cs>	Noun case
<n_cnt>…</n_cnt>	Countability of noun
<n_cmp>…</n_cmp>	Complement of noun
<n_lxc>…</n_lxc>	Lexis
	VERB
<v_inf>…</v_inf>	Verb inflection
<v_agr>…</v_agr>	Subject-verb disagreement
<v_fml>…</v_fml>	Verb form
<v_tns>…</v_tns>	Verb tense
<v_asp>…</v_asp>	Verb aspect
<v_vo>…</v_vo>	Verb voice
<v_fin>…</v_fin>	Usage of finite/infinite verb
<v_ng>…</v_ng>	Verb negation
<v_qst>…</v_qst>	Question
<v_cmp>…</v_cmp>	Complement of verb
<v_lxc>…</v_lxc>	Lexis
	MODAL VERB
<mo_lxc>…</mo_lxc>	Lexis
	ADJECTIVE
<aj_inf>…</aj_inf>	Adjective inflection
<aj_us>…</aj_us>	Usage of positive/comparative/superlative of adjective
<aj_num>…</aj_num>	Number of adjective
<aj_agr>…</aj_agr>	Number disagreement of adjective
<aj_qnt>…</aj_qnt>	Quantitative adjective
<aj_cmp>…</aj_cmp>	Complement of adjective
<aj_lxc>…</aj_lxc>	Lexis
	ADVERB
<av_inf>…</av_inf>	Adverb inflection
<av_us>…</av_us>	Usage of positive/comparative/superlative of adverb
<av_lxc>…</av_lxc>	Lexis
	PREPOSITION
<prp_cmp>…</prp_cmp>	Complement of preposition
<prp_lxc1>…</prp_lxc1>	Normal preposition
<prp_lxc2>…</prp_lxc2>	Dependent preposition
	ARTICLE
<at>…</at>	Article
	PRONOUN
<pn_inf>…</pn_inf>	Pronoun inflection
<pn_agr>…</pn_agr>	Number/sex disagreement of pronoun
<pn_cs>…</pn_cs>	Pronoun case
<pn_lxc>…</pn_lxc>	Lexis

CONJUNCTION	
<con_lxc>…</con_lxc>	Lexis
RELATIVE PRONOUN	
<rel_cs>…</rel_cs>	Case of relative pronoun
<rel_lxc>…</rel_lxc>	Lexis
INTERROGATIVE	
<itr_lxc>…</itr_lxc>	Lexis
OTHERS	
<o_je>…</o_je>	Japanese English
<o_lxc>…</o_lxc>	Collocation
<o_odr>…</o_odr>	Misordering of words
<o_uk>…</o_uk>	Unknown type errors
<o_uit>…</o_uit>	Unintelligible utterance

2.5 Subcorpora

We have also compiled two subcorpora for comparison. One is a native English speakers' corpus and the other is a back-translation corpus. The native English speakers' corpus is considered to be quite useful for comparing the utterances of native speakers and Japanese learners. We were able to make this comparison by collecting the speech data of native speakers, conducting a similar type of interview to that of the SST. The back-translation corpus was compiled mainly by guessing what the learners intended to say in the interview, and then translating this into correct Japanese. With the back-translation corpus, we were able to study how L1 (Japanese) transfer interferes with second language acquisition, or the kinds of things which are difficult for Japanese learners to express in English. As stated above, we performed error tagging only for grammatical and lexical errors. These subcorpora may cover what we are unable to examine solely by error tagging.

3 Automatic error detection

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 an experiment on automatic error detection in which we applied natural language processing (NLP) techniques by using error tag information. We will examine to what extent this could be accomplished using our learner corpus, by describing a method of detecting learners' grammatical and lexical errors and using other techniques that improve the accuracy of error detection with a limited amount of training data.

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 parts, 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 (Figure 3, Method A). During this process, we also determined 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 that we have added one more category (*+1*), namely 'There is no missing word.' (labeled 'C') to the *N* error categories (Figure 3, Method B).

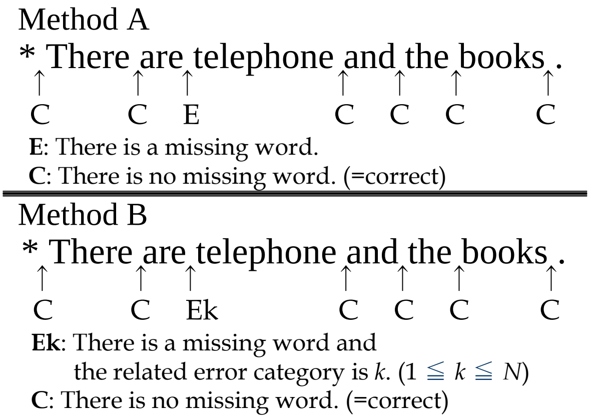

Method A
* There are telephone and the books .
C C E C C C C
E: There is a missing word.
C: There is no missing word. (=correct)
Method B
* There are telephone and the books .
C C Ek C C C C
Ek: There is a missing word and the related error category is k. (1 ≦ k ≦ N)
C: There is no missing word. (=correct)

Figure 3: Detection of omission-type errors

To perform the estimation, we refer to 23 pieces of information as described in Figure 4. 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 Figure 2, '*t*' and '*e*' in '*telephone*'). The word classes and root forms are obtained using 'TreeTagger' (Schmid 1994).

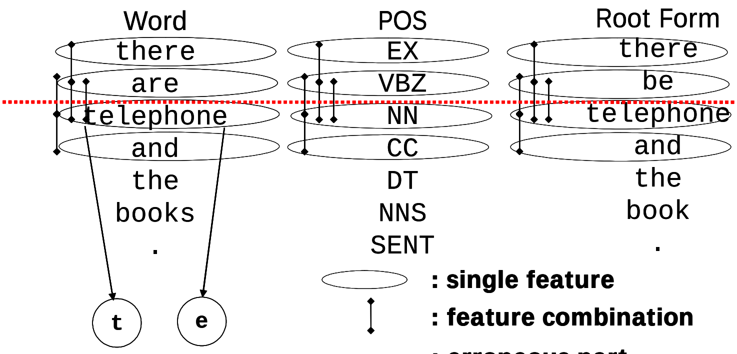


Figure 4: Features used for detecting omission-type errors

3.1.3 Detection of replacement-type/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 Figure 5. 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 and Marcus 1995).

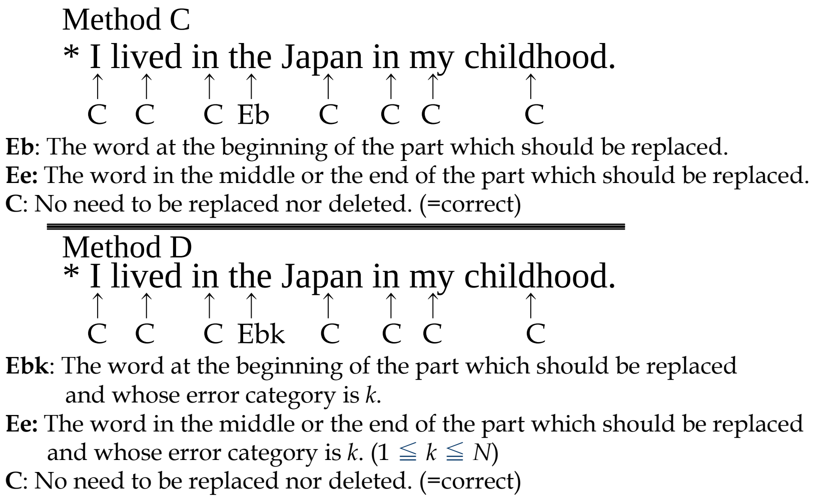

Method C
* I lived in the Japan in my childhood.
C C C Eb C C C C
Eb: The word at the beginning of the part which should be replaced.
Ee: The word in the middle or the end of the part which should be replaced.
C: No need to be replaced nor deleted. (=correct)
Method D
* I lived in the Japan in my childhood.
C C C Ebk C C C C
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)

Figure 5: Detection of replacement/insertion-type errors

To estimate an error category, we refer to 32 pieces of information, as shown in Figure 6. 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 the 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.

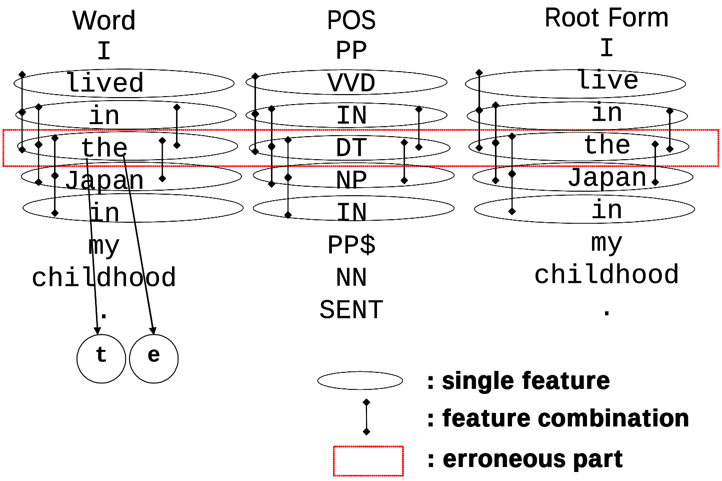


Figure 6: Features used for detecting replacement/insertion type errors

3.1.4 Use of machine learning model

We considered error detection as similar to text categorization in which the goal is, according to Manning and Schutze (1999:575), to classify the topic or theme of a document. Our first attempt was to apply the machine learning model to our framework. We chose the Maximum Entropy (ME) (Jaynes 1957, 1979:15) model, which is used variously to solve text categorization problems and which 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 Figure 7, 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.

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

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

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

Figure 7: 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 Table 3, we selected 13 error categories for detection. We assumed that these errors were more frequent than other errors, and could be identified relatively easily from the context.

Table 3: Error categories to be detected

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

3.2.2 Experiment based on tagged data

We obtained 166 error-tagged transcripts from the SST Corpus. We used 150 files as training data, and 16 files 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 (Figure 8). The best results were obtained for article errors, which were the most frequently occurring errors, as shown in Figure 9:

Article errors		
Omission-type	Recall rate	104/221*100 = 47.06%
	Precision rate	104/162*100 = 64.20%
Replacement/Insertion-type	Recall rate	19/80*100 = 23.75%
	Precision rate	19/56*100 = 33.93%

Figure 8: Recall/precision for the detection of all errors

All errors		
Omission-type	Recall rate	106/300*100 = 35.33%
	Precision rate	106/176*100 = 60.23%
Replacement/Insertion-type	Recall rate	36/590*100 = 6.10%
	Precision rate	36/165*100 = 21.82%

Figure 9: 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 Addition of correct sentences

We added the correct sentences of the following three types: the first type is the native speakers’ speech data subcorpus; the second type is the interviewers’ utterances; the third type is the corrected sentences extracted from the error-tagged data. As for the second type, we added the interviewers' utterances in the entire corpus data (totaling 1,200 transcripts) to the training data. As for the third type, 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 50 error-tagged files. We added a total of approximately 105,000 new correct sentences.

By doing this, the rates of recall and precision in the detection of omission-type errors in all error categories improved by eleven percent and ten percent, respectively. The result remained steady for the detection of replacement and insertion-type errors (Figure 10):

All errors		
Omission-type	Recall rate	72/300*100 = 24.00 (%)
	Precision rate	72/102*100 = 70.59 (%)
Replacement/Insertion-type	Recall rate	36/590*100 = 6.10 (%)
	Precision rate	36/165*100 = 21.82 (%)

Figure 10: Recall/precision for the detection of all errors

For article errors, the recall of detecting omission-type errors decreased by 19 percent, but the precision went up by 4 percent. The precision of the detection of replacement and insertion-type errors increased sharply to 68 percent (Figure 11):

Article errors		
Omission-type	Recall rate	64/221*100 = 28.96 (%)
	Precision rate	64/94*100 = 68.09 (%)
Replacement/Insertion-type	Recall rate	13/80*100 = 16.25 (%)
	Precision rate	13/19*100 = 68.42 (%)

Figure 11: 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 Addition of sentences with 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 Section 3.2.3, and 7,558 sentences that contained artificially made errors, are shown in Figures12 and 13:

Article errors		
Omission-type	Recall rate	136/221*100 = 61.54 (%)
	Precision rate	136/174*100 = 78.16 (%)
Replacement/Insertion-type	Recall rate	20/80*100 = 25.00 (%)
	Precision rate	20/34*100 = 58.82 (%)

Figure 12: Recall/precision for the detection of all errors

All errors		
Omission-type	Recall rate	137/300*100 = 45.67 (%)
	Precision rate	137/181*100 = 75.69 (%)
Replacement/Insertion-type	Recall rate	48/590*100 = 8.14 (%)
	Precision rate	48/154*100 = 31.17 (%)

Figure 13: Recall/precision for the detection of article errors

We obtained a better recall and precision rate for all types of errors. 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 the global context more thoroughly.

3.3 Summary of results

By using the corpus, in its original form, our experiment showed the recall of article errors to be approximately 35 percent and the precision to be approximately 48 percent. By adding corrected sentences and artificially-made errors, recall and precision improved to 43 percent and 68 percent, 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 all kinds of errors. 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 Other possible applications

Other than for the automatic error detection of learners' errors, we assume that the SST Corpus can be utilized in various ways, from fundamental research on second language acquisition, or the design of ELT materials including learners' dictionaries, to the development of a computer-assisted language learning system.

Several research studies have been conducted on the order in which learners acquire various linguistic phenomena, such as negation, tense and aspect. Most studies are examined based on data inducing errors on a particular kind of grammatical rule. This is a good way to obtain the results that the researchers have intended, from a small amount of data. However, results extracted from more spontaneous and large-scale data are more reliable, although it could sometimes be difficult to formulate the extracted results.

It is useful to identify learners' developmental patterns when teachers are planning class activities. For example, teachers can decide which new vocabulary entries to teach to their students by investigating the vocabulary frequency in English of learners who are at a similar level to the students in question. By extracting common errors, teachers can consider which grammatical rules are more difficult to acquire. This will lead to class activities being focused on correcting particular errors. It is also possible to develop improved teaching materials or learners' dictionaries, by using information obtained from learners' developmental patterns. Furthermore, there are several projects being conducted on the development of a computer-assisted language learning (CALL) system by integrating learner corpora and NLP technology. Error analysis, based on the error tagged texts, helps to develop an error diagnostic system, and this will enable the construction of a CALL system that can accept learners' poorly-formed texts and provide them with feedback.

We believe that these applications can be employed effectively by using our corpus which is not only divided into nine proficiency levels, but which also contains rich information on learners' errors.

5 Conclusion

In this paper, we have presented an overview of the SST Corpus, by explaining data collection procedures such as transcribing and tagging, including error tagging for error analysis. 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, Edwin T. 1957. Information theory and statistical mechanics. *Physical Review* 106: 620–630.

Jaynes, Edwin T. 1979. Where do we stand on maximum entropy? In R.D. Levine and M. Tribus (eds.). *The maximum entropy formalism*, 15–118. Cambridge: M.I.T Press.

Manning, Christopher D. and Hinrich Schutze. 1999. *Foundation of statistical natural language processing*. Cambridge: M.I.T Press.

Minnen, Guido, Francis Bond and Ann Copestake. 2000. Memory-based learning for article generation. In *Proceedings of CoNLL-2000 and LLL-2000*: 43–48.

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

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*: 45–49.

Tono, Yukio. 2001. The Standard Speaking Test (SST) Corpus: A 1 million-word spoken corpus of Japanese learners of English and its implications for L2 lexicography. In *Proceedings of ASIALEX2001*: 257–262.