Recognition of English Utterances with Grammatical and Lexical Mistakes for Dialogue-based CALL System

Akinori Ito[1], *Ryohei Tsutsui*[1], *Shozo Makino*[1] *and Motoyuki Suzuki*[2]

[1]Graduate School of Engineering, Tohoku University, Japan
[2]Institute of Technology and Science, The University of Tokushima, Japan
{aito,tutui,makino}@makino.ecei.tohoku.ac.jp, moto@m.ieice.org

Abstract

Our goal is to develop a voice-interactive CALL system which enables language learners to practice words, phrases, and grammars interactively. Such a system must be able to recognize learner's utterances correctly. To enable the recognition of utterances containing grammatical mistakes, we used an n-gram language model trained from generated text. The proposed model achieved recognition performance similar to that of a language model based on a finite-state automaton and manual error rules. We then introduced two error correction techniques to improve recognition performance. One method used the Levenshtein distance between the target sentence and the recognized sentence. The other method used an error-corrective model based on POS n-gram features. The experimental results showed that both methods were able to improve recognition performance.
Index Terms: CALL system, speech recognition, grammatical errors, text generation, error-corrective language model

1. Introduction

Computer-Assisted Language Learning (CALL) systems are now under active development. Most commercial CALL systems are for training in either reading, writing, or listening. Recently, CALL systems for speaking practice have been also developed by exploiting speech recognition technology. Such systems enable learners to exercise pronunciation and intonation [1].

To improve conversation skills, it is important to practice using words or expressions in a real dialogue in addition to completing pronunciation exercises. Several voice-interactive CALL systems aiming to realize a CALL system for conversation practice have been developed [2, 3, 4].

A voice-interactive CALL system must correctly recognize a learner's utterance, including any grammatical mistakes contained in that utterance. A couple of past studies have achieved speech recognition, but the precision of the recognition was still poor because conventional speech recognizers assumed that the utterances to be recognized had no grammatical mistakes. To improve recognition accuracy, we have to enhance the language model (LM) so that utterances with grammatical errors can be recognized correctly.

Language modeling for non-native speech recognition greatly depends on the task of the system. Ehsani *et al.* developed an interactive CALL system for learning Japanese[2]. They used a large number of task-dependent texts to train an n-gram LM. Alternatively, Abe *et al.*[3] and Kweon *et al.* [4] used language models based on a finite state automaton (FSA). An FSA-based LM is suitable for small-sized tasks, where a system developer can describe all possible variations in the utterances. As an FSA-based LM has to accept utterances that contain grammatical mistakes, Abe *et al.* used an FSA, built into which were grammatical mistakes that learners tend to make. Kweon *et al.*, however, combined an FSA without grammatical mistakes with "grammatical error rules" that described predictable mistakes, thereby generating an FSA that accepted utterances with grammatical mistakes.

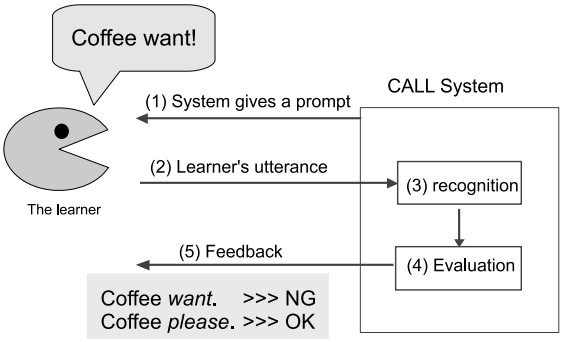

Figure 1: Interactive CALL system overview

N-gram-based LMs are powerful and flexible with regard to the unpredictable mistakes, but they require a large training corpus, which is not always available. Conversely, an FSA-based LMs are easy to develop, but they cannot accept utterances with unpredictable mistakes. It is also difficult for FSA-based LMs to account for the probability of mistakes.

We are currently developing an English-language interactive CALL system for native speakers of Japanese. In this paper, we describe our development of a language modeling method for effectively recognizing non-native speech for interactive CALL systems. Our model is based on an n-gram LM trained by a generated text corpus that accounts for grammatical mistakes.

2. A Voice-Interactive CALL System

Figure 1 shows an overview of an interactive CALL system. Such a system first gives a prompt to a learner. Here, the system plays a certain role in the conversation. The learner then speaks. The utterance is recognized by the system, and the system generates an answer. At the same time, if the utterance contains any mistakes, the system points them and demonstrates a correct version of the utterance.

Generally speaking, it is not easy for learners to converse with computers. In this study, we assumed that the learners would first receive lessons (pre-exercises) on the vocabulary and

grammatical expressions used in specific situations, and then converse with the CALL system on the topics. Pre-exercises make it easier for learners to produce speech when using the system. In addition, assuming a pre-exercise before the dialogue session with the system had the effect of suppressing out-of-task utterances by the learners [4].

As we assumed a pre-exercise before the conversation with the CALL system, we expected learners to respond to the system using the same expressions as those appearing in the pre-exercise. We refer to such a sentence, a correct sentence expected to be uttered by a learner, *a target sentence*. In a real session, however, not all user utterances match the target sentences. We refer to a sentence actually uttered by a learner as *an uttered sentence*. An uttered sentence often contains grammatical and lexical mistakes.

The utterances are recognized by the system using a speech recognizer. The recognition results are different from the uttered sentences on account of recognition errors. We refer to a sentence obtained from the speech recognizer as a *recognized sentence*. We now have three sentence types for a single utterance – the target sentence, the uttered sentence and the recognized sentence. The existence of a target sentence is the biggest difference between this system and usual speech recognition tasks.

3. Recognition of Utterances Using Generated N-gram

3.1. Overview

In this section, we describe the language modeling used in our system. As we mentioned before, either n-gram LM or FSA-based LM can be used to recognize utterances in conventional interactive CALL systems. The creation of an n-gram-based LM requires a great deal of training data, but it is difficult to gather a sufficient amount of data for non-native utterances in specific task domains. Conversely, an FSA-based LM is easier to develop, but it is less flexible than an n-gram LM and it cannot accept utterances with unpredictable mistakes.

To address this problem, we trained an n-gram model using artificial sentences generated from the target sentence and grammatical error rules. Systems using artificial text data were first developed for the rapid domain adaptation of spoken dialog systems [6]. We combined the LM training using artificial texts with grammatical error rules that were used in the FSA-based methods [4].

The difference between our task and ordinary speech recognition is the existence of target sentences in our task. While some variation (including grammatical and lexical mistakes) is possible, we expected the uttered sentences to be similar to the corresponding target sentences. Therefore, we used the target sentences as "seeds" for generating training sentences, and we applied the error rules to the target sentences to generate mistake variations.

Figure 2 shows an overview of the n-gram training. We first applied error rules in a certain probabilities to generate sentences with mistakes. Here, we used two kinds of error rules: the generic error rules and the corpus-based error rules. We then trained an n-gram using the generated sentences.

3.2. Error rules

As we described above, we used two kinds of error rules. The corpus-based error rules were extracted from a transcription of

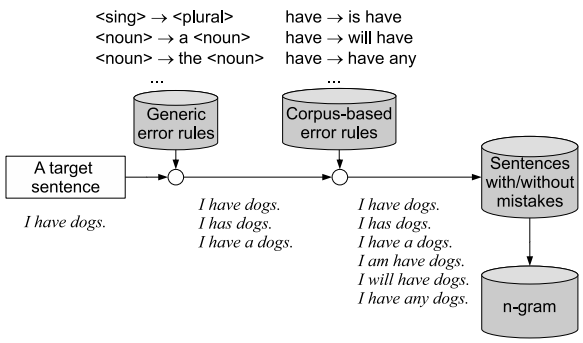

Figure 2: Training of an n-gram from generated text

English utterances by native Japanese speakers. We exploited the SST (Standard Speaking Test) corpus[7], which consists of transcriptions of interviews with native Japanese speakers in English. The grammatical and lexical errors in this corpus are manually annotated with the correct expressions. Table 1 shows the mistakes most frequently observed in the corpus. The symbol ε denotes the absence of a corresponding word. The error types SUB, DEL and INS indicate a substitution error, a deletion error and an insertion error, respectively. These results show that most of the mistakes made by native Japanese speakers involve articles. From these mistakes, we used only substitution and deletion error rules as corpus-based error rules. We did not use insertion errors because contextual information is indispensable to the simulation of such errors. For example, we would need rules such as "the word *the* is inserted between a preposition and a noun," rather than "the word *the* is inserted between two words." Otherwise, we would generate sentences such as "*What a time a the is the it?*" which can hardly be expected to be uttered, even by novice English learners. Thus, we have to incorporate the context of the mistake into the error rule. However, it is not useful to use word-level context (e.g. *want to play golf → want to the play golf*) because the small data size prevents us from covering all variations in word context. Therefore, we did not use insertion errors extracted from the corpus. Instead, most insertion errors were covered by the generic error rules described below.

The generic error rules were manually created to cover a wider range of mistakes. These error rules include the insertion of articles, confusion between singular and plural, and verb tense mistakes. As POS tags were needed to apply these error rules to specific sentences, we used Brill's Tagger[8] to tag the target sentences.

On applying these two kinds of rules, we first applied the generic error rules, followed by the corpus-based rules. The probabilities of choosing each corpus-based rule were in proportion to the frequencies of the corresponding mistake types in the SST corpus. Conversely, the probabilities of applying the generic error rules were given *a priori*. We conducted a preliminary experiment and confirmed that the final recognition performance was not greatly affected by the probabilities of the generic error rules.

3.3. Experiment

We carried out an experiment to confirm the performance of an n-gram-based LM. The test data consisted of 33 English sentences spoken by 11 male native Japanese speakers. The utterances were recorded in the following manner: First, the

Table 1: Grammatical mistakes in the SST corpus

Rank	Freq.	Correct	Uttered	Type
1	1227	a	ε	DEL
2	896	the	ε	DEL
3	403	ε	the	INS
4	299	a	the	SUB
5	207	to	ε	DEL
6	148	ε	a	INS
7	139	is	ε	DEL
8	126	in	ε	DEL
9	118	the	a	SUB
10	115	an	ε	DEL

Table 2: Experimental conditions

Acoustic model	512-mixture 5-state HMM
Acoustic feature parameters	MFCC, ΔMFCC, ΔΔMFCC, Δpow,ΔΔpow
Training data	ERJ database[5] About 100,000 words uttered by 100 males
Decoder	Julius 3.5.3 (n-gram) / Julian 3.5.3 (FSA)
Test data	11 males, 3 sentences/speaker
Error rate of uttered sentences with respect to the target sentences	26.9%

Japanese sentences and the corresponding English sentences were given to the speaker. The speaker attempted to memorize the sentences for a few minutes and then repeated these sentences to the system without looking at the English guide (he was still allowed to refer to the Japanese sentences). As Japanese sentences were presented before each utterance, no off-task utterances were observed. Table 2 shows the experimental conditions.

Our results showed a 25.3% WER using n-gram trained from the generated text, while the WER of an FSA with manual error rules was 25.4%. The performance of the n-gram-based LM was almost to the same as that of the conventional method, wherein the developer has to write the FSA and error rules manually.

4. Error Correction of Recognition Results

4.1. Problems observed in the recognition results

While n-gram-based LM gave a WER similar to that of the FSA-based LM, the recognition results still contained sentences that were completely different from the uttered sentences, as in the following example:

target	we	are	a	little	early$\cdots$
uttered	we	are		little	early$\cdots$
recognized	will	her		be	very$\cdots$

These results were caused by a poor LM. To correct such errors, we tried to introduce two error correction techniques. The first one used the Levenshtein distance between the recognized and target sentences. The second one used an error-corrective model based on POS frequency features.

In both approaches, we first generated the n-best recognition candidates and then rescored the candidates by combining the speech recognition score with a score for error correction. Let $s_1, \ldots, s_N$ be recognition candidates, and $X(s_k)$ be a recognition score of s_k calculated by the decoder. By using an error correction score $Q(s_k)$, we calculate a new score

$$X'(s_k) = X(s_k) + \beta Q(s_k) \tag{1}$$

where β is a combination factor. Finally we selected a candidate s_K such that $X(s_K) \geq X(s_k)$ for $1 \leq k \leq N$.

4.2. Error correction using Levenshtein distance from the target sentence

As the speakers utter an English sentences referring to corresponding Japanese sentences, we did not expect the uttered sentences to be very different from the target sentences. Therefore, we used the Levenshtein distance between s_k and the target sentence τ as an error correction score. Let the Levenshtein distance of two sentences be $d(s, s')$. Then we used

$$Q(s_k) = -d(\tau, s_k). \tag{2}$$

4.3. Error-corrective model using POS features

Another technique for correcting errors is the error-corrective model[9]. This model first calculates the vectors of n-gram features $\boldsymbol{\phi}(s_k)$. We used POS unigram, POS bigram and POS trigram frequencies as these features, which produced good performance even when the size of the training data was small[10]. For example, when we had *noun, article, adjective,..,* as the parts of speech, $\boldsymbol{\phi}(s_k)$ looks like this:

$$\boldsymbol{\phi}(s_k) = \begin{bmatrix} N(noun; s_k) \\ N(article; s_k) \\ N(adjevtive; s_k) \\ \vdots \\ N(noun, noun; s_k) \\ N(noun, article; s_k) \\ \vdots \\ N(noun, noun, noun; s_k) \\ N(noun, noun, article; s_k) \\ \vdots \end{bmatrix} \tag{3}$$

where $N(p_1, p_2, \ldots; s)$ stands for the occurrence frequency of the POS sequence $p_1, p_2, \ldots$ in the sentence s.

A score was then calculated using a weight vector $\boldsymbol{\alpha}$ as

$$Q(s_k) = \boldsymbol{\alpha}^T \boldsymbol{\phi}(s_k) \tag{4}$$

where $\boldsymbol{\alpha}$ is a vector of the same dimension as $\boldsymbol{\phi}(s_k)$. Using the perceptron algorithm[9], the weight vector $\boldsymbol{\alpha}$ was trained to give a higher score to the low-WER candidates.

We needed both positive and negative examples to train the model. We used the candidate with the lowest WER as a positive example and the 10 candidates with the highest WERs as negative examples for a single utterance. We used the 300 best candidates for each utterance, and used 301 utterances spoken by seven males as the training data.

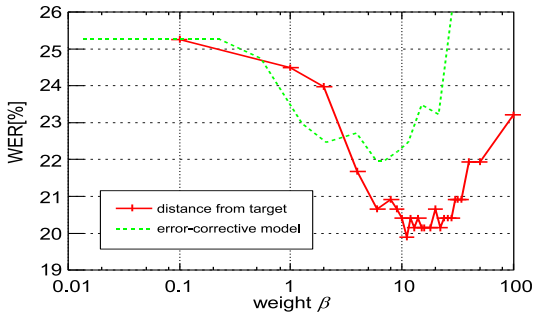

Figure 3: WER results using the proposed methods

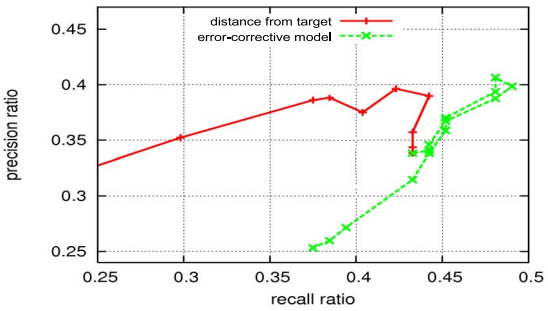

Figure 4: Recall and precision results using the proposed methods

4.4. Experimental results

We first calculated the 1000 best candidates for each utterance, and then rescored them using the two error correction methods. The experimental conditions are same as shown in Table 2. Figure 3 shows the WER results obtained using the two above-mentioned methods. These results showed that both methods were able to improve the WER over speech recognition conducted without error correction (the result for smaller β). The Levenshtein-distance-based method was better than the error-corrective model. However, the method based on the error-corrective model has the advantage that it can be applied without any the target sentences.

Next, we evaluated the same results from the aspect of recall and precision of mistake detection. Mistake detection was conducted by aligning the target sentence with the recognized sentence. For example, one substitution error, one insertion error and two deletion errors were detected in the following example.

target	we	are	a	little	early	
recognized	we	were			early	little
error		SUB	DEL	DEL		INS

The detected errors were compared with the results of error detection results conducted on the uttered sentence itself. A detection result was regarded as a misdetection when any of the following were incorrect: word position, recognized word, or type of mistake.

The relationships between the recall and precision rate are shown in Figure 4. These results showed that the error-corrective model produced a better recall ratio than did the method based on the Levenshtein distance.

5. Conclusions

We have developed a method to recognize non-native English utterances with the intention of creating an interactive CALL system. First, we applied error rules to generate sentences with mistakes. We then trained an n-gram model using these sentences. An n-gram model trained with the generated sentences performed recognition with similar accuracy to the manually-created FSA. Next, we applied two error correction methods to the recognition results in order to improve the recognition performance. The first was based on the distance between the recognition candidate and the target sentence. The second was an error-corrective model using POS n-gram features. While the distance-based method produced a lower WER, the feature-based method gave a better recall ratio than did the distance-based method.

Though we were able to improve the WER and recall/precision rate, the absolute performance of the method is still too low to allow its application to an actual CALL system. We must further improve the performance of the system by using more training data and improving the error rules.

6. Acknowledgements

This work is partially supported by a Grant-in-Aid for Scientific Research (Grant No. 20320075) from Japanese Society for Promotion of Science.

7. References

[1] F. Ehsani and E. Knodt, “Speech technology in computer-aided language learning: strengths and limitations of a new CALL paradigm,” Language Learning & Technology, vol. 2, no. 1, pp. 45-60 (1998)

[2] F. Ehsani, J. Bernstein and A. Najmi: “An interactive dialog system for learning Japanese”, Speech Communication, vol. 30, no. 2–3, pp.167-177(2000)

[3] K. Abe, K. Tanaka, T. Kawahara, M. Shimizu and M. Dantsuji, “A study on the speech recognition performance for the interactive system for English learning,” Proc. Acoust. Soc. Jpn. Spring Meeting, 2-5-30, pp. 113–114 (2002)

[4] O.-P. Kweon, A. Ito, M. Suzuki and S. Makino, “A Grammatical Error Detection Method for Dialogue-based CALL system”, Journal of Natural Language Processing, vol. 12, no. 4, pp.137-156 (2005)

[5] N. Minematsu, Y. Tomiyama, K. Yoshimoto, K. Shimizu, S. Nakagawa, M. Dantsuji and S. Makino, “Developmemt of English Speech Database Read by Japanese to Support CALL Research,” Proc. Int. Cong. on Acoustics, vol. I, pp. 557–560 (2004)

[6] M. Akbacak, Y. Gao, L. Gu and H.-K. J. Kuo, “Rapid Transition to New Spoken Dialogue Domains: Language Model Training Using Knowledge from Previous Domain Applications andWeb Text Resources,” Proc. Interspeech, pp. 1873–1876 (2005)

[7] Y. Tono, T. Kaneko, H. Isahara, T. Saiga, and E. Izumi, “The Standard Speaking Test Corpus: A 1 million-word spoken corpus of Japanese learners of English and its implications for L2 lexicography,” Proc. 2nd Asialex International Congress, pp. 257–262, 2001.

[8] E. Brill, “A Simple Rule-Based Part of Speech Tagger,” Proc. ANLP-92, 3rd Conf. on Applied Natural Language Processing, pp.152-155, 1992.

[9] B. Roark, M. Saraclar and M. Collins, “Corrective Language Modelling for Large Vocabulary ASR with the Perceptron Algorighm”, Proc. ICASSP, vol.1, pp.749-752(2004)

[10] T. Oba, T. Hori and A. Nakamura, “An Approach to Efficient Generation of High-Accuracy and Compact Error-Corrective Models for Speech Recognition,” Proc. Interspeech, pp. 1753–1756 (2007)