SEAMLESS ERROR CORRECTION INTERFACE FOR VOICE WORD PROCESSOR

Junhwi Choi , Kyungduk Kim, Sungjin Lee,
Seokhwan Kim, Donghyeon Lee, Injae Lee, Gary Geunbae Lee

Department of Computer Science and Engineering,
Pohang University of Science and Technology, Pohang,Gyungbuk,Korea
{chasunee, getta, junion, megaup, semko, lij1984, gblee}@postech.ac.kr

ABSTRACT

In this paper, we propose an error correction interface for a voice word processor. This correction interface includes user intention understanding and automatic error region detection. For accurate correction, we include a confirmation process that includes an error region control command and a re-uttering command. We evaluate the performance of the user intention understanding first, and we evaluate the effectiveness of our interface compare to a general two-step error correction interface.

Index Terms— Error Correction, Voice Word Processor

1. INTRODUCTION

A voice word processor is an automatic speech recognition (ASR) system that translates wave signals into text. Even when the ASR system has a low error rate, the recognized results frequently include error words. To perfect a document, an error correction process is required. The correction process can be performed by selecting an erroneous portion of the text using a keyboard, a mouse, or other devices and speaking replacement text. However, in some usage scenarios, error correction using only voice commands is required. A handicapped person who cannot use either arm may want the error correction to use only voice. In addition, users initially tend to try to correct misrecognized results using their own speech [1] and often remain in the same speech modality even when faced with repeated recognition errors [2]. Therefore, error correction using only voice commands may also be convenient for non-handicapped users.

In general, voice-only error correction is a two-step process. In the first step, the users speak a portion of the recognized text to select a target position to correct. Next, the users speak a replacement text. These two steps can perform one correction. However, as McNair and Waibel [3] suggest, the correction process can instead be performed in a single step. In one-step correction, users speak only their replacement text, and the system automatically recognizes it correctly and finds the error region to replace.

In this paper, we propose a seamless error correction interface for a voice word processor. Seamless error correction is processed like one-step error correction, without any explicit command to enter the correction mode. Our interface automatically understands the purpose of the utterance whether the intention is to type a new sentence or to correct a misrecognized sentence. Then, the system detects an error region and corrects it. To complement the understanding of user intention, our interface provides a confirmation process. To demonstrate the effectiveness of our interface, we evaluate it by comparison to a general two-step process.

2. RELATED WORK

For accurate and seamless error correction, the system should achieve three functionalities: First, the replacement utterance should be recognized accurately. The replacement text that users speak is not a full sentence, but a sub-sentence. Because the language models for ASR are usually adapted to full sentences, the replacement utterance may not be correctly recognized. Therefore, an alternative language model is required. For this task, Vertanen and Kristensson [4] developed a flexible merge model that improved accuracy by combining information from the original recognition with information from the spoken correction. Second, the error region should be accurately identified. For the second task, Vertanen and Kristensson [5] presented three new models for automatically aligning the error region and the correction: a 1-best model, a word confusion network model, and a revision model. Third, the system should understand the purpose of the utterance. This third task deals with entering the correction mode. This functionality should precede the previous two tasks, as those tasks can be done after entering the correction mode. The previous two tasks do not address the third task because they assume that the correction mode has already been entered. This paper concentrates on the user intention understanding process.

In commercial products, Nuance Dragon Naturally Speaking Solution [6] provides a voice interface for word processors. The interface follows a general two-step process for editing sentences and correcting errors. First, users utter continuously supported commands and the portion of already typed sentences that they intend to edit or correct. Then, the system assigns numbers to all of the detected regions. The users select a region to edit, and the command is applied to the region. In correction, there is one more process. The system recommends several candidates to replace. The user then select a recommendation or utter the right sentence. We will compare our seamless error correction with the general two-step error correction process.

3. SEAMLESS ERROR CORRECTION

Fig. 1 shows the word processing workflow using our interface. After the user utters the sentences to type or correct, our system detects analysis regions for accurate understanding of intention. In this process, the system finds the region of previously typed sentences most similar to the current utterance by the local alignment of the pronunciation sequences. Considering the characteristics of ASR, even a misrecognized sentence has a

similar pronunciation sequence to the sentence that the users really want to type. Furthermore, for purposes of correction, Vertanen and Kristensson [5] observed that, without explicit instruction, users tend to speak correctly recognized words surrounding an error region. The better the performance of the ASR system, the more similar the pronunciation sequences are. After that, user intention understanding proceeds, that is, the classification of correction or non-correction. When the intention of the current utterance is for correction, detected error region is replaced automatically. Otherwise, the current utterance is inserted at the end of the document. If the user finds some errors in the intention understanding process, the user can use confirmation process.

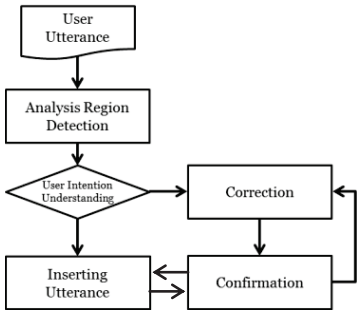

Figure. 1. Workflow of Seamless Error Correction Interface

In confirmation process, our interface provides four commands: error region window control command, re-uttering command, user intention changing command, and cancel command. Users can adjust error region with the error region window control command. If correction utterance is recognized with error but user intention is classified correctly, user can re-utter. The user intention changing command changes correction to non-correction.

4. USER INTENTION UNDERSTANDING

The key novel process in our interface is user intention understanding. User intention understanding can be accomplished by the observation of clear speech [7][8]. User utterances to ASR usually have the characteristics of clear speech, which is a speaking style adopted by a speaker aiming to increase the intelligibility for a listener. To make their speech more intelligible, users will make on-line adjustments; typically, they will speak slowly and loudly, and they will articulate in a more exaggerated manner [7][8]. Furthermore, the utterances for correction display these characteristics more conspicuously than the utterances for non-correction.

We approach the task as a classification problem. We collect data from users, label the data with intentions, and extract and refine some of the data's features.

4.1. Wizard of OZ Data Collection

To classify the user's intention for the current utterance, we should collect user utterances and label them as correction or non-correction. Therefore, we collected data using the Wizard of OZ (WOZ) method, behind which there is a human supervisor [9]. The supervisor imitates the operation of our seamless error correction interface.

Each user was required to create a document using the system. The user was guided regarding how to correct misrecognized results in ASR. In this process, we did not impose any prosodic characteristics on the user, and thus, we could collect data with the natural prosody of speech for correction or non-correction.

The supervisor prepared 4~5 misrecognized sentences from ASR for each task sentence. These prepared misrecognized sentences had actually occurred in ASR, as the presented situation should be a realistic representation of using the seamless error correction interface. The prepared sentences evenly included insertion errors, substitution errors, and deletion errors. Then, the supervisor listened to the user's utterances from behind the system. The supervisor showed the user a misrecognized result on purpose to elicit a correction utterance from the user. Then, the supervisor replaced the error with the correct sentence.

We recorded all utterances and labeled them with labels indicating whether the intention of the utterance was for correction or for non-correction. We collected 458 utterances (211 for non-correction, 247 for correction) from 10 users.

We refined the raw wave data to training data to use machine-learning techniques. The training data consist of several instances. Each instance contains the features below and is labeled with the user's intention by human annotators. However, some features vary with each user and utterance, so we constructed the training data with a focus on normalizing those features.

4.2. Normalization of Prosodic Features

First, to classify user intention, we focus on prosodic features. Usually, prosodic features are used to classify clear speech [7][8].

For all users and utterances, the prosodic features should be normalized to the ratio of the features of a target utterance to those of the current utterance, where the target utterance is the utterance of a previously typed sentence.

Calculating the ratios of utterances of whole typed sentences to the current utterance may cause difficulty in understanding the intention. For more accurate modeling, we should calculate the ratio of a target sub-utterance, which users really want to replace, to the correction utterance. Therefore, the task of finding the target sub-utterance to be replaced should be performed first. Therefore, our interface begins by detecting the analysis region. Then, it calculates the ratio between the target and the current utterance. The equation is as follows:

$$(Feature)\ Ratio = \frac{(Feature)\ of\ T}{(Feature)\ of\ C}$$

where T is the most similar target sub-utterance and C is the current utterance.

The ratio value represents the direction of change and also represents the degree of change from the features of the target sub-utterance to the features of the current utterance; therefore, it may produce confusion because the degree of change depends on each user. Therefore, we also use tendency. Tendency represents only the direction of change. The equation is as follows:

$$(Feature)\ Tendency = \begin{cases} 0 & (Feature)\ Ratio > 1 \\ 1 & else \end{cases}$$

This equation means that if the feature of a current utterance is larger than that of a target sub-utterance, the tendency value is 1.

We have separated the specific features into three categories. Table 1, below, presents the normalized prosodic features we use.

Table. 1. Normalized Prosodic Features for Classifying User Intention

Prosodic Category	Specific Features
Normalized Pitch	Ratio of Max Pitch
	Ratio of Min Pitch
	Tendency of Max Pitch
	Tendency of Min Pitch
Normalized Intensity	Ratio of Max Intensity
	Ratio of Min Intensity
	Tendency of Max Intensity
	Tendency of Min Intensity
Normalized Duration	Ratio of Full Utterance Duration per Syllable
	Ratio of Sub-utterance Duration per Syllable
	Tendency of Full Utterance Duration per Syllable
	Tendency of Sub-utterance Duration per Syllable

4.3. Distance of Pronunciation Sequences

We measure a distance between the pronunciation sequence of the most similar target sub-utterance and the pronunciation sequence of the current utterance. We use a Levenshtein distance as a distance measure, but it depends on the length of the pronunciation sequence. Even when the different proportions of two pronunciation sequences are equal, the shorter pronunciation sequence has a lower Levenshtein distance value, and therefore, it must be normalized. The equation is below.

$$Distance = \frac{LD}{l}$$

where LD is the Levenshtein distance value between the pronunciation sequence of the target sub-utterance and that of the current utterance, and l is the length of the pronunciation sequence of the current utterance. The normalized value reflects the difference between the pronunciation sequences of the current utterance and the target sub-utterance. The lower the value, the more similar the pronunciation sequences between the current utterance and the target sub-utterance.

4.4. Feature Verification

We generated training data that included 458 instances from raw wave data. We labeled 211 instances as non-correction and the other 247 instances as correction. Each instance has an intention label and 13 features (12 normalized prosodic features and 1 distance feature).

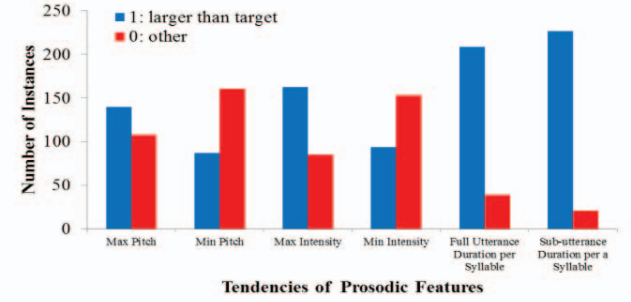

Figure. 2. Distribution of correction instances by the tendencies of prosodic features

Fig. 2 shows the distributions of correction instances by the tendencies of their prosodic features. The best-separated prosodic feature is the tendency of sub-utterance duration per character. We can see that the correction utterances tend to be slower than the target utterances. We can also find characteristics of clear speech, in that the pitch range and intensity range of the correction utterances are slightly wider than those of the target sub-utterances.

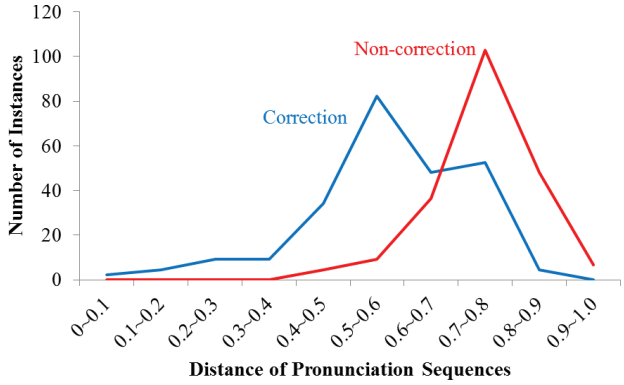

Figure. 3. Distribution of instances by the distance of pronunciation sequences

Fig. 3 shows a distribution of instances by the distance of their pronunciation sequences. Correction utterances are distributed in a relatively low region of the distance value, separated from non-correction utterances, so that the distance feature is effective for classifying intention.

5. EXPERIMENT

To evaluate the performance of the user intention understanding of our interface, we used a support vector machine as a classifier with radial basis function (RBF) and we validated our approach with a 10-fold cross validation. As a baseline, we used majority of correction.

We also evaluated the effectiveness of our interface compared to a general two-step error correction interface that we developed as a baseline. We gave 8 users a task making documents consisting of 10 sentences with 264 Korean syllables.

5.1. Performance of User Intention Understanding

In voice word processor, the false correction and non-correction cause duplicated tasks making users uncomfortable. Therefore, accuracy is the most important evaluation value.

Table 2 shows the validation results. The best result was obtained by combining the tendencies of all prosodic features and the distance of pronunciation sequences; it achieved 82.91% classification accuracy. The ratio features reflected information on both the direction and the degree of change. As single normalized prosodic features, the ratio features were effective. However, in a combined model (all normalized prosodic features + distance feature), we found that the degree of change confused the classification user intention, because the degree of change was relatively smaller than the distance; therefore, in the combined model, considering only the tendency features was more effective.

The most effective single feature was the distance of pronunciation sequences; it achieved 78.89% classification

Table. 2. User Intention Classification Accuracy

Features Category	Accuracy (%)
Baseline (majority)	54.27
Normalized pitch	75.37
Normalized intensity	75.88
Normalized duration per syllable	77.89
All normalized prosodic features	79.90
Distance	78.89
All normalized prosodic features + distance	79.90
Normalized prosodic features without ratios + distance	**82.91**

accuracy, but it depended on the performance of the ASR. ASR with a high error rate produced many errors and caused a higher distance feature. Therefore, other features are required that are independent of the performance of ASR.

5.2. Effectiveness of Seamless Error Correction

We measured average syllables and average turns to complete a task including command syllables and turns. We also measured average syllables and average turns to correct an error including command syllables and turns.

Table. 3. Effectiveness of Two Error Correction Interfaces

	Syllables	Turns	Syllables per error	Turns per error
Two-step error correction interface	392.23	21.82	29.67	3.45
Seamless error correction interface	346.35	16.35	21.54	2.25

Table 3 shows the effectiveness of two error correction interfaces. Average syllables per task were reduced about 45.88 and average turns per task were also reduced about 5.47; the differences were statistically significant ($p = 0.028$ for syllables, $p = 0.018$ for turns; paired T-test). Our interface improved the effectiveness about 13.25% for average syllables per task and about 33.46% for average turns per task. For an error correction, a general two-step error correction interface needed about 29.67 syllables and about 3.45 turns, but our interface needed about 21.54 syllables and about 2.25 turns; the differences were also statistically significant ($p = 0.002$ for syllables per error , $p = 0.036$ for turns per error; paired T-test). In an error correction, our interface also improved the effectiveness about 37.74% for syllables and about 53.33% for turns. This result shows that our interface can work effectively with 82.91% classification accuracy for user intention understanding.

6. CONCLUSION

In this paper, we observe the characteristics of correction utterance alongside the characteristics of clear speech. We classify user intention using the normalized prosodic features and the distance feature and achieve 82.91% success. Furthermore, with 82.91% success of user intention understanding, we prove the effectiveness of our seamless error correction by comparison to a general two-step error correction.

By understanding a user's intention, we are able to provide a system that can automatically enter correction mode with only the correction utterance of a replacement text, that is, a seamless correction. With this seamless correction, in composing a document, users need not remember any voice commands for entering correction mode and may simply speak sentences they want to type. Therefore, the efficiency of the process may be increased. In addition, the developers of a voice word processor need not design any voice commands for entering the correction mode.

7. ACKNOWLEDEGMENT

This work was supported by the Quality of Life Technology (QoLT) development program, 10036458, Development of voice word-processor and voice-controlled computer software for physical handicapped person funded by the Ministry of Knowledge Economy (MKE, Korea).
This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0027953).

8. REFERENCES

[1] Sharon Oviatt and Robert Van Gent, "Error resolution during multimodal human-computer interaction," in *Proc. International Conference on Spoken Language Processing*, 1996, pp. 204-207.
[2] Christine A. Halverson, Daniel B. Horn, Clare-Marie Karat, and John Karat, "The beauty of errors: Patterns of error correction in desktop speech systems," in *Proc. INTERACT*, 1999, pp. 133–140.
[3] Arthur E. McNair and Alex Waibel, "Improving recognizer acceptance through robust, natural speech repair," in *Proc. International Conference on Spoken Language Processing*, 1994.
[4] Keith Vertanen and Per Ola Kristensson, "Getting it Right the Second Time: Recognition of Spoken Corrections," in *SLT '10: Proc. the 3rd IEEE Workshop on Spoken Language Technology*, 2010.
[5] Keith Vertanen and Per Ola Kristensson, "Automatic selection of recognition errors by respeaking the intended text," in *ASRU `09: IEEE Workshop on Automatic Speech Recognition and Understanding*, December 2009, pp. 133-140.
[6] http://nuance.com
[7] Seung-Jae Moon and Björn Lindblom, "Interaction between duration, context, and speaking style in English stressed vowels," in *Journal of the Acoustical Society of America*, 1994, Volume 96, Issue 1, pp. 40-55.
[8] Rajka Smiljanić and Ann R. Bradlow, "Production and perception of clear speech in Croatian and English," in *Journal of the Acoustical Society of America*, 2005, Volume 118, Issue 3, pp. 1677-1688.
[9] John F. Kelley, "CAL – A Natural Language program developed with the OZ Paradigm: Implications for Supercomputing Systems," in *Proc. 1st International Conference on Supercomputing Systems*, ACM, 1985, pp. 238–248