論文 / 著書情報
Article / Book Information

題目(和文)	携帯端末における音声認識のための効率的な誤り訂正
Title(English)	Efficient Error Correction for Mobile Speech Recognition
著者(和文)	梁原
Author(English)	Yuan Liang
出典(和文)	学位:博士（学術）, 学位授与機関:東京工業大学, 報告番号:甲第10018号, 授与年月日:2015年9月25日, 学位の種別:課程博士, 審査員:篠田　浩一,亀井　宏行,徳永　健伸,室田　真男,村田　剛志,藤井　敦
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第10018号, Conferred date:2015/9/25, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

（博士課程）
Doctoral Program

論 文 要 旨

THESIS SUMMARY

専攻： Department of	Computer Science 専攻	申請学位（専攻分野）： Academic Degree Requested	博士 （ Philosophy ） Doctor of
学生氏名： Student's Name	Yuan Liang	指導教員（主）： Academic Advisor(main)	Koichi Shinoda
		指導教員（副）： Academic Advisor(sub)	

要旨（英文 800 語程度）
Thesis Summary（approx.800 English Words ）

The dissertation "Efficient Error Correction for Mobile Speech Recognition" consists of 7 chapters.

Chapter 1 [Introduction] presents the background and the motivation of error correction for mobile speech recognition. The goal of our research is given, that is to design a simple multimodal error correction interface and develop an efficient candidates generating algorithm for users to correct speech recognition errors with less effort in the speech input applications.

Chapter 2 [Automatic Speech Recognition (ASR) System] briefly introduces the statistical modeling approach for ASR, the architecture of an ASR system and its five main components: feature extraction, acoustic model (AM), language model (LM), pronunciation dictionary, and decoding. We show the most common AMs, which are GMM-HMM AMs, and the most common LMs, that are the N-gram LMs. The procedure of generating a WCN and outputting the 1-best posterior decoding result is also explained. Two most common performance metrics used to evaluate an ASR system are introduced, that are word error rate (WER) and word accuracy (Acc). Three error types in ASR are also shown in this chapter, which are insertion, substitution, and deletion error.

Chapter 3 [Error Correction for Speech Recognition: A Review] presents the previous researches in the field of error correction for speech recognition. WCN has been widely used to generate a candidate list for a speech recognition error. We review several researches use user validated prefix in error correction to re-evaluate the succeeding transcription and to re-order the WCN. The reason why previous researches did not use the entire user validated suffix is explained. The problems of using the Web text data in augmenting the training data for adapting a LM or in rescoring the first pass speech recognition result are shown in this chapter. We provide a practical, feasible way of using Google's Web N-gram corpora in our proposed LCM to get the candidates, which are complementary to the candidates got from WCN. Interface evaluation metrics used in Human-computer interaction (HCI) such as the Keystroke level model (KLM) and the Enhancing KLM are introduced. Interface evaluation metrics used in speech error correction such as the number of users' actions are also shown here.

Chapter 4 [Gesture Based Error Correction Interface] describes the proposed gesture

based error correction interface. First, three gestures are defined to correct errors when using our interface. Second, an example error correction procedure using our interface is shown in this chapter. Third, how to correct successive multiple errors by using our interface is explained.

Chapter 5 [Long Context Match (LCM) Method for Generating Candidate Word List] explains the proposed LCM algorithm for realizing the interface. We suppose users mark misrecognized words from left to right first and then correct them also from left to right, therefore we can assume all the preceding words of the misrecognized word are correct or corrected by users, we call this information user validated prefix. Since the succeeding words of the misrecognized word may contain error words, we call the correct succeeding words user validated suffix. We develop LCM by using not only the user validated prefix, but also the user validated suffix and Web text data to get the candidates complementary to the WCN. For each error word, the system generates the candidate list for its corresponding correct word by using LCM in the following steps: Step-1) extracts its location, error type, and user-validated prefix and suffix to make search queries; Step-2) uses those queries to search higher-order N-grams for matching word sequences, and from each of them extracts the word in its position as a candidate; Step-3) if the number of candidates is one, the system directly outputs it. Otherwise, go to Step 4; Step-4) calculates its LM score and AM score for each candidate; Step-5) calculates its posterior probability of each candidate, and orders candidates in their descending order. We further propose an algorithm for generating accurate candidates by using a combination of LCM and WCN ("LCM + WCN").

Chapter 6 [Experiments] describes the setup of the baseline ASR system, the way of collecting ASR errors, and the Corpus of Spontaneous Japanese (CSJ) corpus we use for our research. Comparing with the conventional WCN method, the proposed "LCM + WCN" method improved the 1-best accuracy by 23% and improved the MRR by 28%. Comparing with the conventional WCN-based interface, our "LCM + WCN"-based interface successfully reduced the user's load by 12%. We confirm the effectiveness of using user validated suffix in error correction. Comparing with the "LCM (pre) + WCN", the proposed "LCM (pre + suf) + WCN" improved the 1-best accuracy by 8.4% and improved the MRR by 2.9%. Comparing with the "LCM (pre) + WCN"-based conventional interface, our "LCM (pre + suf) + WCN"-based proposed interface successfully reduced the user's load by 12%.

Chapter 7 [Conclusions and Future Work] ends this thesis by drawing conclusions and giving some suggestions for the future work. The future work of focusing on interface implementation and usability testing is described.