SEMANTIC-ORIENTED ERROR CORRECTION FOR SPOKEN QUERY PROCESSING

Minwoo Jeong[1], Byeongchang Kim[2], Gary Geunbae Lee[1]

[1]Department of Computer Science and Engineering, POSTECH, Pohang, Korea
{stardust, gblee}@postech.ac.kr
[2]Division of Computer and Multimedia Engineering, UIDUK University, Gyeongju, Korea
bckim@uiduk.ac.kr

ABSTRACT

Voice input is often required in many new application environments such as telephone-based information retrieval, car navigation systems, and user-friendly interfaces, but the low success rate of speech recognition makes it difficult to extend its application to new fields. Popular approaches to increase the accuracy of the recognition rate have been researched by post-processing of the recognition results, but previous approaches were mainly lexical-oriented ones in post error correction. We suggest a new semantic-oriented approach which can correct semantic level errors as well as lexical errors, and is more accurate for especially domain-specific speech error correction. Through extensive experiments using a speech-driven in-vehicle telematics information application, we demonstrate the better performance of our approach and some advantages over previous lexical-oriented approaches.

1. INTRODUCTION

New application environments such as telephone-based retrieval, car navigation systems, and mobile information retrieval, require voice interface in processing user queries. In these environments, keyboard input is inconvenient or sometimes impossible because of the spatial limitations on mobile devices and the instability in manipulation of the device.

However, due to the low recognition rate of the current speech recognition systems, in order to increase the accuracy of information retrieval, appropriate post-processing is required such as post error correction. The average recognition accuracy of the representative continuous speech recognition system is not more than 90% in the word-based and at most 70% in the sentence-based in a practical situation [1].

Most previous researches in post-processing have been based on statistical methods utilizing the probabilistic information of words spoken in a speech dialogue situation, and the language models adapted to the application [2, 3]. The performance of such systems depends on the size and quality of speech recognition result, or on the database of collected error strings since they are directly dependent on the lexical items.

They use the probability of mistakenly recognized words, the co-occurrence information extracted from the words and their neighboring words, and tagged word bi-grams, which are all lexical clues in error strings. Such approaches based on lexical information of words have shown some successful results, but they still have major drawbacks. The error patterns constructed are available but are not abundant, because it costs much expense to collect them; so, there are many cases where they fail to recover the original strings from the lexical error patterns. Also, since they are so sensitive to the error patterns, it occasionally happens to mistakenly recognize a correct word as an error word.

We suggest a more robust semantic-oriented error correction approach, which is an improvement over but easily integrated into the previous fragile lexical-based approaches. In our approach, in addition to the lexical information, we use high level syntactic and semantic information of the words in the speech transcription. We obtain semantic information from some knowledge base such as general thesauri and a special dictionary which we construct by ourselves to contain some domain specific knowledge to the target application.

Our semantic-oriented approach has some advantage over the lexical based ones, since it is less sensitive to each error pattern. Also, the approach has more broad coverage over error patterns, since several similar common error strings in semantic ground can be reduced to one semantic error pattern, which enables us to improve the probability of recovering from erroneous recognition results.

2. RELATED WORKS

2.1. Spoken query processing

The major problem of speech-driven information retrieval (IR) and question answering (QA) is the decreasing of performance due to the recognition errors in query sentences by automatic speech recognition (ASR) systems. Erroneous queries drop the precision and recall of IR and QA system. Some authors investigated the relation of ASR errors and precision of IR [4, 5]. They evaluated the effectiveness of IR system through various error rates of 35 queries of TREC. Their researches show that increasing word error rate (WER) quickly decreases the precision of IR.

Another group investigated the performance of spoken queries in NTCIR collections [6]. They evaluated a variety of speakers, and calculated error rate with respect to a query term, which is a keyword used for the retrieval. They showed that error rate of query terms was generally higher than WER irrespective of the speakers. In other words, recognition of content words related on to the IR or QA performance was more difficult than that of normal words.

A method to improve precision of speech-driven IR is introduced by the other group [1]. They suggested a new type of IR system tightly-integrated with a speech input interface. In their system, the document collection provided an adaptation of the language model of the ASR, which results in drop of the word error rate. Through the same framework, they also tried to solve the out-of-vocabulary (OOV) word problem, which is another important issue in the ASR and IR integration.

2.2. Post error correction

There are some previous works for reducing the errors of speech recognition. Ringger and Allen [2] suggested the noisy channel model for speech error correction. They simplified a statistical machine translation (MT) model called IBM statistical MT model [7]. They tried to construct a general post-processor that can correct errors generated by any speech recognizer. The model consists of two parts: a channel model, which accounts for errors made by the ASR, and the language model, which accounts for the likelihood of a sequence of words being uttered in the first place.

They trained the channel model and the language model both using some transcriptions from TRAINS-95 dialogue system which is a train traveling planning system [8]. Here, the channel model has the distribution that an original word may be recognized as an erroneous word.

Kaki et al. [3] suggested another approach, that is, a straightforward and intuitive method to robustly handle many kinds of recognition errors. They collected many error patterns that occurred in a speech translation system, and constructed also a corpus consisting of a general word string from that domain. They could correct any type of errors by matching the strings in the transcription with error patterns in the database. However, their approach has a disadvantage in that they are only feasible to the trained (or collected) error patterns, hence if the domain of the application is changed, the system must be trained again from the start, which is time and money consuming.

In some similar areas such as spelling error correction or optical character recognition (OCR) error correction, NLP researchers traditionally identified five levels of errors in text: (1) a lexical level, (2) a syntactic level, (3) a semantic level, (4) a discourse structure level, and (5) a pragmatic level [9]. In spelling correction and OCR error correction problem, correction schemes mainly focused on non-word errors in lexical level. However, errors of speech recognition are real-word errors which should be classified into syntactic and semantic level errors, because a recognizer produces word sequences existing in lexicon.

The previous works focused on lexical level error correction, thus may not be appropriate to be applied to different speakers and environments. These works commonly depend on a large amount of training corpus for the error correction model and the language model. So, previous approaches require fluent results of ASR and are dependent on specific speakers and environments. On the other hand, our method takes far less training corpus, and it is possible to implement the method easily and in a short time to obtain the same or better error correction rate because it utilizes the semantic information of the application domain.

3. SEMANTIC-ORIENTED ERROR CORRECTION APPROACH

This section presents a semantic-oriented approach to correct erroneous outputs of a speech recognizer using domain knowledge. We focus on real-word error detection and correction, so our approach is based on the domain knowledge for performing semantic level error detection and correction.

3.1. Lexico-semantic pattern

A lexico-semantic pattern (LSP) is a structure where linguistic entries and semantic types are used in combination to abstract certain sequences of the words in

a text. It has been used in the area of natural language interface for database (NLIDB) [10] and a TREC QA system for the purpose of matching the user query with the appropriate answer types [11, 12].

In an LSP, linguistic entries consist of words, phrases and part-of-speech (POS) tags, such as 'YMCA,' 'Young Men's Christian Association,' and 'NNP[1].' Semantic types consist of common semantic classes and domain-specific (or user-defined) semantic classes. The common semantic tags again include attribute-values in databases, such as '@corp' for company name like 'IBM,' and predefine 83 semantic category values, such as '@location' for location name like 'New York' [10]. Figure 1 shows an example of predefined common semantic category values.

@a_lang	@event	@magazine	@person	@unit_area
@action	@family	@mammal	@phenomenon	@unit_count
@artifact	@fish	@month	@planet	@unit_date
@belief	@food	@mountain	@plant	@unit_length
@bird	@game	@movie	@position	@unit_money
@book	@god	@music	@reptile	@unit_power
@building	@group	@nationality	@school	@unit_rate
@city	@language	@nature	@season	@unit_size
@color	@living_thing	@newspaper	@sports	@unit_speed
@company	@location	@ocean	@state	@unit_temperature
@continent		@organization	@status	@unit_time
@country	@exam		@subject_area	@unit_volume
@date	@hobby	@method	@substance	@unit_weight
@direction	@law	@address	@team	@unit_age
@disease	@level	@appliance	@transport	
@drug	@living_part	@art	@weekday	
		@computer	@picture	
		@course	@river	
		@deed	@room	
			@sex	

Figure 1. Common semantic category values

In domain-specific application, well defined semantic concepts are required, and the domain-specific semantic classes represent these requirements. The domain-specific semantic classes include special attribute names in databases, such as '%action' for 'active' and 'inactive,' and semantic category names, such as '%hobby' for 'reading' and 'recreation,' for which the user wants specific meaning in the application domain. Moreover, we used the classes to abstract out several synonyms into a single concept. For example, a domain-specific semantic class '%question' represents some words, such as 'question', 'questions', 'asking', and 'answer.'

The words in a query sentence are converted into the LSP through two steps. First, a morphological analysis is performed, which segments a sentence of words into morphemes, and adds POS tags into morphemes [13]. Next, each morpheme of the sentence is converted into a suitable semantic symbol by searching several types in the semantic dictionaries.

[1] Part-of-speech tag denoting a proper noun used in Penn TreeBank.

3.2. Construction of domain knowledge

Because Fujii et al. [1] have shown the importance of the language model which well describes the domain knowledge, we reflect the domain information with template database: LSP-converted queries of the source statements which are used for the actual error detection and correction task after speech recognition. The template queries are automatically acquired by the Query-to-LSP translation from the source statements using two semantic category dictionaries: domain dictionary and ontology dictionary.

The domain dictionary is a subset of the general semantic category dictionary, and focuses only on the narrow extent of the knowledge it concerns, since it is impossible to cover all the knowledge of the world in implementing an application. On the other hand, the ontology dictionary reflects the pure general knowledge of the world; hence it performs a supplementary role in extracting semantic information.

The domain dictionary provides the specific vocabulary which is used at semantic representation task of a user query and the template database. Assuming that some speech statements for a specific target domain is predefined, a record of the template database is made up of a fixed number of LSP elements, such as POS tags, semantic tags, and domain-specific semantic classes. Table 1 shows example of template abstracted by LSP converting in a predefined domain of education.

Phrases	LSP
Reading trainer	%hobby @position
Fairy tale trainer	
Fairy tale oral narrator	
Recreation coach	

Table 1. Example of template abstracted by LSP

For semantic-oriented error correction, we constructed a domain knowledge, which consists of domain dictionary, thesaurus, and template queries that are similar to question types in QA system. Query sentences are semantically abstracted by LSP's, and are collected for the template database.

Query-to-LSP transforms given query into corresponding LSP, and the LSP's enhance the coverage of extraction by information abstraction through many-to-one mapping between queries and an LSP. The transformation consists of two phases: Named entity

(NE) recognition and NE tagging [14]. NE recognition discovers all the possible semantic types for each word by consulting a domain dictionary and a thesaurus. When a semantic type for a given word does not exist in the domain dictionary, we attempt to discover the semantic types using the thesaurus. NE tagging selects a semantic type for each word so that a sentence can be mapped into our LSP sequence.

3.3. Semantic-oriented correction process

Now we will show the working mechanism of post error correction of a speech recognition result using the domain knowledge of template database and domain-specific dictionary. Figure 2 is a schematic diagram of the post error correction process.

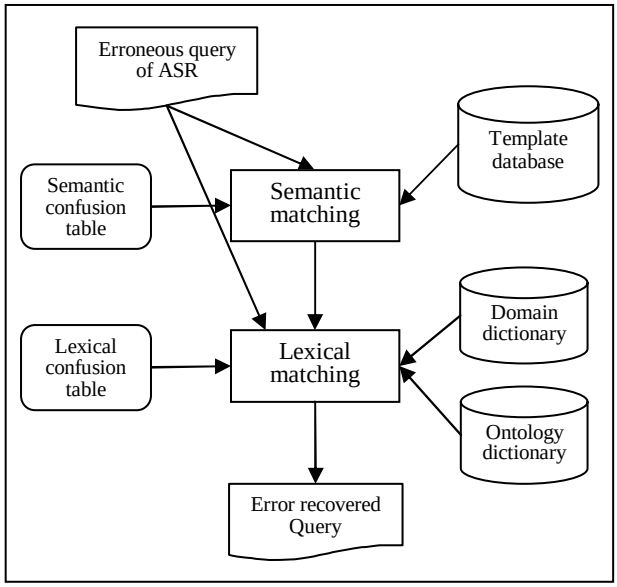

Figure 2. Semantic-oriented Error Correction Process

The overall process is divided into two stages: semantic recovery and lexical recovery stage. In semantic recovery stage, recognized query is converted into the corresponding LSP. The converted LSP may be ill-formed depending on the recognized query. Semantic recovery is performed by replacing these syntactic or semantic errors using a semantic confusion table. We used pre-collected template database to recover the semantic level errors, and the technique for searching most similar templates are based on minimum edit-distance dynamic search, which has been used as a similarity search in many areas such as spelling correction, OCR post correction, and DNA sequence analysis [15]. The semantic confusion table provides the matching cost to the dynamic programming search process. The 'minimum edit distance' between two words is originally defined as the minimum number of deletions, insertions, and substitutions required for transforming one word into the other. We compute the minimum edit distances between the erroneous LSP's and the templates in the database using the similar cost functions at the LSP level, and select, as the final template LSP, the one which has the minimum distance among them.

After this procedure, lexical recovery is performed in the next stage. Recovered semantic tags and the erroneous query produced by ASR are the clue of lexical recovery. Erroneous query and recovered LSP template are aligned by dynamic programming again, and then some candidates are discovered as the most similar words to the original input words in the domain dictionary or ontology dictionary. As many lexical candidates under the same semantic class may exist, we select the most similar one as the final correction word using the minimum edit distance on the lexical level. In minimum edit distance, cost function is important to the sensitive matching, so we use a confusion table of lexical variation at this stage.

Figure 3 shows an example of semantic error correction process using the same data in TRAIN-95 [2].

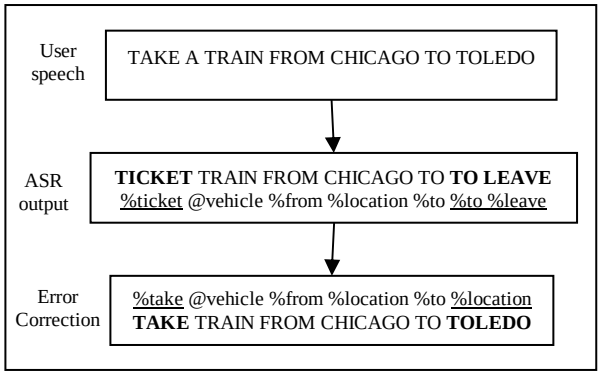

Figure 3. Example of semantic-oriented error correction

4. EXPERIMENTS

We performed several experiments on the domain of in-vehicle telematics IR related to navigation question answering services. The speech transcripts used in the experiments were composed of 462 queries, which were collected in real application. We used two Korean speech recognizers: a speech recognizer made by LG-Elite (LG Electronics Institute of Technology) and a Korean commercial speech recognizer, ByVoice (refer to http://www.voicetech.co.kr).

We implemented the previous noisy channel model (NC-Model) to compare with our system [2]. We generated an NC-Model which consists of a trigram

language model, and a channel model based on IBM model 4 [7]. We adopted fertility probabilities from IBM model 4, but discarded the distortion probabilities, because distortion didn't appear in speech recognition. We used open source toolkit for generating the models: SLM toolkit for language model [16] and GIZA toolkit for channel model [17].

We divided the 462 queries into the training set and the test set, and evaluated the results of 6-cross-validation for both our model and the NC-Model. For our post-processor, we constructed 436 semantic templates for evaluating our system, and 3,195 entries of domain dictionary.

Table 2 presents the experiments results of word accuracies of baseline ASR, NC-Model, and our post-processor. The performances of baseline systems were about 79% ~ 81% on the utterances in in-vehicle telematics IR domain.

	ASR A	ASR B
Baseline	79.51%	81.25%
NC-Model	85.79%	84.57%
our system	87.03%	86.93%

Table 2. Experiment of word accuracy

This result shows that the post-processing of speech recognition result is a viable approach to improve the performance. Using both baseline ASR systems, we achieved 7.5% and 5.7% increase of word accuracy. In comparison with NC-Model, our system achieves 1.3% and 2.4% of more accurate error correction performance in this domain.

We also evaluated term error rate (TER), which is more important factor of performance in IR, that is, an error rate of content words directly related to the performance of IR and QA system. Table 3 shows the result of the experiments for WER and TER.

	ASR A		ASR B	
	WER	TER	WER	TER
Baseline	20.49%	56.17%	18.75%	64.14%
NC-Model	14.21%	23.32%	15.43%	30.03%
our system	12.97%	19.34%	13.07%	27.02%

Table 3. Comparison of word error rate and term error rate

According to the comparison of WER and TER, baseline ASR systems alone are not appropriate to process the user's queries in speech-driven IR or QA systems. However, with post error correction, error reduction rate of TER is much higher than that of WER. With this result, our semantic-oriented error correction method is more appropriate in speech-driven IR and QA applications. Comparing with the NC-Model that has been the best approach in the error correction so far, our semantic-oriented error correction shows other successful possibilities for speech recognition error correction.

5. CONCLUSION

We proposed a semantic-oriented approach in the speech recognition error correction which shows better performance in domain-specific IR applications. Our approach has the following advantages: First, it is fast and easy to develop, and leads to computationally simple implementation. The background knowledge ontology dictionaries are independent of the speech recognition lexicon, and open-vocabulary, and are constructed only once, except for the domain dictionary which depends on a specific application domain, but is very small compared to the ontology dictionary.

Second, because the LSP scheme transforms pure lexical entries into abstract semantic categories, the size of the error pattern database can be reduced remarkably, and it also increases the coverage and robustness compared with the previous pure lexical entries which can only deal with the morphological variants.

Third, with all these facts, the LSP correction has a high possibility of generating semantically correct correction due to the massive use of semantic contexts. Hence, it shows a high performance, especially when combined with domain-specific speech-driven natural language IR and QA systems.

6. ACKNOWLEDGMENTS

This work was supported by mid-term strategic funding (MOCIE, ITEP). We thank LG Electronics Institute of Technology for providing their speech recognition results.

7. REFERENCES

[1] Atsushi Fujii, Katunobu Itou, and Tetsuya Ishikawa, "A method for open-vocabulary speech-driven text retrieval," in p*roceeding of the 2002 conference on Empirical Methods in Natural Language Processing*, pp.188-195, 2002.

[2] Eric K. Ringger and James F. Allen, "A fertility model for post correction of continuous speech recognition," *ICSLP'96*, pp. 897-900, 1996.

[3] Satoshi Kaki, Eiichiro Sumita, and Hitoshi Iida, "A Method for Correcting Speech Recognition Using the Statistical

features of Character Co-occurrence," *COLING-ACL'98*, pp.653-657, 1998.

[4] J. Barnett, S. Anderson, J. Broglio, M. Singh, R. Hudson, and S.W. Kuo, "Experiments in spoken queries for documents retrieval," *in proceedings of Eurospeech, volume 3*, pages 1323-1326, 1997.

[5] F. Crestani, "Word recognition errors and relevance feedback in spoken query processing," *in proceedings of the 2000 Flexible Query Answering Systems Conference*, pages 267-281, 2000

[6] Atsushi Fujii, Katunobu Itou, Tetsuya Ishikawa, "Speech-driven text retrieval: Using target IR collections for statistical language model adaptation in speech recognition," in A. R. Cohen, E. W. Brown, and S. Srinivasan, editors, *Information Retrieval Techniques for Speech Application (LNCS 2273)*, pages 94-104, 2002.

[7] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A Statistical Approach to Machine Translation," *Computational Linguistics*, 16(2):79-85, 1990.

[8] James F. Allen, Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski, "A Robust System for Natural Spoken Dialogue," *in proceedings of the 34th Annual Meeting of the ACL*, 1996.

[9] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, 24(4): 377-439, 1992.

[10] Hanmin Jung, Gary Geunbae Lee, Wonseug Choi, KyungKoo Min, and Jungyun Seo, "Multi-lingual question answering with high portability on relational databases," *IEICE transactions on information and systems*, vol E-86D, No2, pp. 306-315, 2003.

[11] Geunbae Lee, Jungyun Seo, Seungwoo Lee, Hanmin Jung, Bong-Hyun Cho, Changki Lee, Byung-Kwan Kwak, Jeongwon Cha, Dongseok Kim, JooHui An, Harksoo Kim, and Kyungsun Kim, "SiteQ: Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP," *in proceedings of the 10th Text Retrieval Conference (TREC-10)*, Washington D.C., 2001.

[12] Haksoo Kim, Kyungsun Kim, Gary Geunbae Lee, and Jungyun Seo, "MAYA: A Fast Question-Answering System Based on a Predictive Answer Indexer," *in proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, Workshop on Open-Domain Question Answering, 2001.

[13] Gary Geunbae Lee, Jeongwon Cha, and Jong-Hyeok Lee, "Syllable pattern-based unknown morpheme segmentation and estimation for hybrid part-of-speech tagging of Korean," *Computational Linguistics*, Vol 28, No 1, pp 53-70, 2002.

[14] Juhui An, Seungwoo Lee, and Gary Geunbae Lee, "Automatic acquisition of Named Entity tagged corpus from World Wide Web," *in proceedings of the 41st annual meeting of the ACL (poster presentation)*, 2003.

[15] Robert A. Wagner and Michael J. Fischer, "The string-to-string correction problem," *Journal of the ACM*, 21(1):168–173, 1974.

[16] R. Rosenfeld, "The CMU Statistical Language Modeling Toolkit and its use in the 1994 ARPA CSR Evaluation," I*n ARPA Spoken Language Technology Workshop*, 1995.

[17] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky, "Statistical machine translation, Technical report," *John Hopkins University Summer Workshop*, 1999.