Using Higher-level Linguistic Knowledge for Speech Recognition Error Correction in a Spoken Q/A Dialog

Minwoo Jeong
Department of Computer
Science and Engineering,
POSTECH, Pohang, Korea
stardust@postech.ac.kr

Byeongchang Kim
Division of Computer and
Multimedia Engineering,
Uiduk University,
Gyeongju, Korea
bckim@uiduk.ac.kr

Gary Geunbae Lee
Department of Computer
Science and Engineering,
POSTECH, Pohang, Korea
gblee@postech.ac.kr

Abstract

Speech interface is often required in many application environments such as telephone-based information retrieval, car navigation systems, and user-friendly interfaces, but the low speech recognition rate makes it difficult to extend its application to new fields. Several approaches to increase the accuracy of the recognition rate have been researched by error correction of the recognition results, but previous approaches were mainly lexical-oriented ones in post error correction. We suggest an improved syllable-based model and a new semantic-oriented approach to correct both semantic and lexical errors, which is also more accurate for especially domain-specific speech error correction. Through extensive experiments using a speech-driven in-vehicle telematics information retrieval, we demonstrate the superior 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, often require speech interface to conveniently process user queries. In these environments, keyboard input is inconvenient or sometimes impossible because of spatial limitation on mobile devices and instability in manipulating the devices.

However, because of the low recognition rate in current speech recognition systems, the performance of speech applications such as speech-driven information retrieval (IR) and question answering (QA), and speech dialogue systems is very low. The performance of the serially connected spoken QA system, based on the QA system from text input which has 76% performance and the output of the ASR which operated at a 30% WER, was only 7% (Harabagiu et al., 2002). (Harabagiu et al., 2002) exposes several fundamental flaws of this simple combination of an automatic speech recognition (ASR) and QA system, including the importance of named entity information, and the inadequacies of current speech recognition technology based on n-gram language models.

The major problem of speech-driven IR and QA is the decreasing of the performance due to the recognition errors in ASR systems. Erroneously recognized spoken queries drop the precision and recall of IR and QA system. Some authors investigated the relation of ASR errors and precision of IR (Barnett et al., 1997; Crestani, 2000). They evaluated the effectiveness of the IR systems through various error rates using 35 queries of TREC. Their researches show that the increasing word error rate (WER) quickly decreases the precision of IR. Another group investigated the performance of spoken queries in NTCIR collections (Fujii et al., 2002A). They evaluated a variety of speakers, and calculated the error rate with respect to a query term, which is a keyword used for the retrieval. They showed that the WER of the query terms was generally higher than that of the general words irrespective of the speakers. In other words, recognition of content words related to the IR and QA performance was more difficult than that of normal words. So, they introduced a method to improve the precision of speech-driven IR by suggesting a new type of IR system tightly-integrated with a speech input interface (Fujii et al., 2002B). In their system, document collection provides an adaptation of the language model of the ASR, which results in a drop of the word error rate.

For this reason, some appropriate adaptation techniques are required for overcoming speech recognition errors such as post error correction. ASR error correction can be one of the domain adaptation techniques to improve the recognition accuracy, and the primary advan-

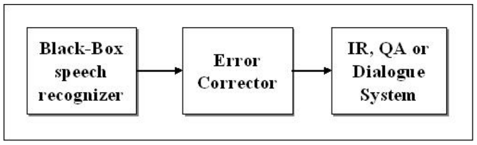


Figure 1: Adaptation via Post Error Correction

tage of the error correction approach is its independence of the specific speech recognizer. If the speech recognizer can be regarded as a black-box, we can perform robust and flexible domain adaptation through the post error correction process. Figure 1 shows the paradigm of this post error correction approach.

One approach in post error correction, which is a straightforward and intuitive method to robustly handle many kinds of recognition errors, was rule-based approach (Kaki et al., 1998). (Kaki et al., 1998) collected many lexical error patterns that occurred in a speech translation system in Japanese. They could correct any type of errors by matching the strings in the transcription with lexical error patterns in the database. However, their approach has a disadvantage in that the correction is only feasible to the trained (or collected) lexical error patterns.

Another approach has been based on a statistical method utilizing the probabilistic information of words in a spoken dialogue situation and the language models adapted to the application domain (Ringger and Allen, 1996). (Ringger and Allen, 1996) applied the noisy channel model to the correction of the errors in speech recognition. They simplified a statistical machine translation (MT) model called an IBM model (Brown et al., 1990), and 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. 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 (Allen et al., 1996). Here, the channel model has the distribution that an original word may be recognized as an erroneous word. They use the probability of mistakenly recognized words, the co-occurrence information extracted from the words and their neighboring words, and the 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 performance of such systems depends on the size and the quality of speech recognition result, or on the database of collected error strings since they are directly dependent on lexical items. The error patterns constructed are available but not enough, because it is expensive to collect them; so in many cases, they fail to recover the original strings from the lexical specific error patterns. Also, since they are sensitive to the error patterns, they occasionally mis-identify a correct word as an error word.

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

In the next section, we first describe a general noisy channel model for ASR error correction and discuss some problems with them. We then introduce our improved channel model especially for Korean language in section 3. We also propose a new high-level error correction model using syntactic and semantic knowledge in section 4. We prove the feasibility of our approach through some experiments in section 5, and draw some conclusions in section 6.

2 Noisy Channel Error Correction Model

The noisy channel error correction framework has been applied to a wide range of problems, such as spelling correction, statistical machine translation, and ASR error correction (Brill and Moore, 2000; Brown et al., 1990; Ringger and Allen, 1996). The key idea of noisy channel model is that we can model some channel properties through estimating the posterior probabilities.

The problem of ASR error correction can be stated in this model as follows: For an input sentence, $O = o_1, o_2, \ldots, o_n$ produced as the output sequence of ASR, find the best word sequence, $\hat{W} = w_1, w_2, \ldots, w_n$, that maximizes the posterior probability $P(W|O)$. Then, applying Bayes' rule and dropping the constant denominator, we can rewrite as:

$$\hat{W} = \arg\max_{W} P(W|O) = \arg\max_{W} P(W)P(O|W) \quad (1)$$

Now, we have a noisy channel model for ASR error correction, with two components, the source model $P(W)$ and the channel model $P(O|W)$. The probability P(W) is given by the language model and can be decomposed as:

$$P(W) = \prod_{i} P(w_i|w_{1,i-1}) \quad (2)$$

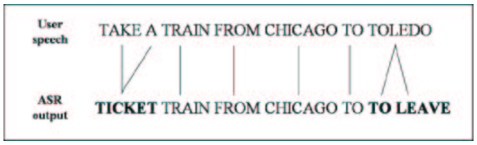

User speech
TAKE A TRAIN FROM CHICAGO TO TOLEDO
ASR output
TICKET TRAIN FROM CHICAGO TO TO LEAVE

Figure 2: Example of Word-based Channel Model

The distribution $P(W)$ can be defined using n-grams, structured language model (Chelba, 1997), or any other tool in the statistical language modeling.

Next, the conditional probability, $P(O|W)$ reflects the channel characteristics of the ASR environment. If we assume that the output word sequence produced under ASR are independent of one another, we have the following formula:

$$P(O|W) = \prod_i P(o_{1,i}|w_{1,i}) = \prod_i P(o_i|w_i) \quad (3)$$

So,

$$\begin{aligned} \hat{W} &= \arg\max_W P(W)P(O|W) \\ &= \arg\max_W (\prod_i P(w_i|w_{1,i-1}) \prod_i P(o_i|w_i)) \quad (4) \end{aligned}$$

However, this simple one-to-one model is not suitable to handling split or merged errors, which frequently appear in an ASR output, because we assume that the output word sequence are independent of one another. For example, [1]figure 2 shows a split or a merged error problem. To solve this problem, Ringger and Allen used the fertility of pre-channel word (Ringger and Allen, 1996). Following (Brown et al., 1990), we refer to the number of post-channel words o_i produced by a pre-channel word w_i as a fertility. They simplified the fertility model of IBM statistical MT model-4, and permitted the fertility within 2 windows such as $P(o_{i-1}, o_i|w_i)$ for two-to-one channel probability, and $P(o_i|w_i, w_{i+1})$ for one-to-two channel probability. So, the fertility model can deal with (TO LEAVE, TOLEDO) substitution. But this improved fertility model only slightly increased the accuracy in experiments (Ringger and Allen, 1996), and we think the major reason is due to the data-sparseness problem. Because substitution probability is based on the whole word-level, this fertility model requires enormous training data. We call the model a word-based channel model, because this model is based on the word-to-word transformation. The word-based model focused on inter-word substitutions, so it requires enough results of ASR and transcription pairs. Considering the cost of building the enough amount of correction pairs, we need a smaller unit than a word for overcoming the data-sparseness.

3 Syllable-based Channel Model

We suggest an improved channel model for smaller training data. If we can use smaller unit such as letter, phoneme or syllable than word, relatively smaller training set is needed. For dealing with intra-word transformation, we suggest a syllable-based channel model, which can deal with syllable-to-syllable transformation. This model is especially reasonable for Korean. In some agglutinative languages such as Korean, syllable is a basic unit of written form like a Chinese character. In Korean, the average number of syllables in one word is about three or four.

3.1 The Model

Suppose $S = s_1, s_2, \ldots, s_n$ is a syllable sequence of ASR output and $W = w_1, w_2, \ldots, w_m$ is a source word sequence, then our purpose is to find the best word sequence $\hat{W}$ as follows:

$$\hat{W} = \arg\max_W P(W|S) \quad (5)$$

We can apply the same Bayes' rule and decompose the syllable-to-word channel model into syllable-to-syllable channel model.

$$\begin{aligned} P(w|s) &= \frac{P(s|w)P(w)}{P(s)} \propto P(s|w)P(w) \\ &\approx P(s|x)P(x|w)P(w) \quad (6) \end{aligned}$$

So, final formula can be written as:

$$\hat{W} = \arg\max_W (P(W)P(X|W)P(S|X)) \quad (7)$$

Here, $P(S|X)$ is the probability of a syllable-to-syllable transformation, where $X = x_1, x_2, \ldots, x_n$ is a source syllable sequence. $P(X|W)$ is a word model, which can convert syllable lattice into word lattice. The conversion can be done efficiently by dictionary look-up.

This model is similar to a standard hidden markov model (HMM) of continuous speech recognition. In speech recognition system, $P(S|X)$ can be an acoustic model in signal-to-phoneme level, and $P(X|W)$ can be a pronunciation dictionary. Then, we applied the fertility into our syllable-to-syllable channel model. We set the maximum 2-fertility of syllable, which was determined experimentally.

3.2 Training the Model

To train the model, we need a training data consisting of $\{X, S\}$ pairs which are manually transcribed strings and ASR outputs. And, we align the pair based on minimizing the edit distance between x_i and s_i by dynamic

[1] This example is from (Ringger and Allen, 1996).

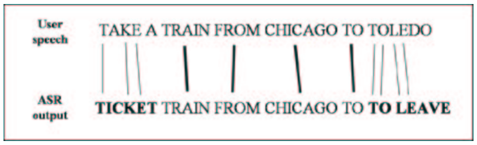

User speech
TAKE A TRAIN FROM CHICAGO TO TOLEDO
ASR output
TICKET TRAIN FROM CHICAGO TO TO LEAVE

Figure 3: Example of Syllable-based Channel Model

programming. [2]Figure 3 shows an alignment for the syllable-model (For understanding, we use an English example and a letter-to-letter alignment. In Korean, each syllable is clearly distinguished much like a letter in English.). For example, (TO LEAVE, TOLEDO) pair in previous section can be divided into (TO, TO), (L, L), (EA, E), and (VE, DO) with fertility 2.

We can then calculate the probability of each substitution $P(s_i|x_i)$ by Maximum-Likelihood Estimation (MLE). Let $C(x_i)$ be the frequency of source syllable, and $C(x_i, s_i)$ be the frequency of events where x_i substitute s_i. Then,

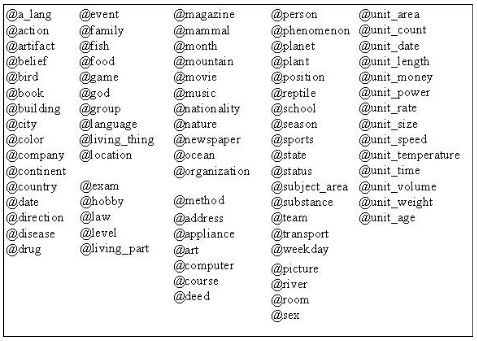
$$P_{MLE}(s_i|x_i) = \frac{C(x_i, s_i)}{C(x_i)} \tag{8}$$

The total number of theoretical unique syllables is about ten thousands in Korean, but the number of syllables, which appeared at least one time, is about 2,300 in a corpus which has about 3 billion syllables. Thus, we used Witten-Bell method for smoothing unseen substitutions (Witten and Bell, 1991). Let $T(x_i)$ be the number of substitution types, and N be the number of syllables in a training data. For Witten-Bell discounting, we should define $Z(x_i)$, which is the number of syllable x_i with count zero. Then, we can write as follows:

$$P_{WB}(s_i|x_i) = \frac{T(x_i)}{Z(x_i)(N + T(x_i))}, if\ C(x_i, s_i) = 0 \tag{9}$$

3.3 Decoding the Model

Given a syllable sequence S, we want to find $\arg\max_W (P(W)P(X|W)P(S|X))$. This will be to return an N-best list of candidates according to the models, and then rescore these candidates by taking into account the language model probabilities. To rescore the candidates, we used Viterbi search algorithm to find the best sequence. For implementation of candidate generation, we store the syllable channel probabilities $P(s_i|x_i)$ as a hash-table to pop them easily and fast. The system can generate a candidate word sequence network using syllable channel model and a lexicon. And then, we can find optimal sequence which has the best probability through Viterbi decoding by including a language model.



@a_lang
@action
@artifact
@belief
@bird
@book
@building
@city
@color
@company
@continent
@country
@date
@direction
@disease
@drug

@event
@family
@fish
@food
@game
@god
@group
@language
@living_thing
@location

@exam
@hobby
@law
@level
@living_part

@magazine
@mammal
@month
@mountain
@movie
@music
@nationality
@nature
@newspaper
@ocean
@organization

@method
@address
@appliance
@art
@computer
@course
@deed

@person
@phenomenon
@planet
@plant
@position
@reptile
@school
@season
@sports
@state
@status
@subject_area
@substance
@team
@transport
@weekday
@picture
@river
@room
@sex

@unit_area
@unit_count
@unit_date
@unit_length
@unit_money
@unit_power
@unit_rate
@unit_size
@unit_speed
@unit_temperature
@unit_time
@unit_volume
@unit_weight
@unit_age

Figure 4: Common semantic category values

4 Using Syntactic and Semantic Knowledge

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 a text: (1) a lexical level, (2) a syntactic level, (3) a semantic level, (4) a discourse structure level, and (5) a pragmatic level (Kukich, 1992). In spelling correction and OCR error correction problem, correction schemes mainly have focused on non-word errors at the lexical level, which is an isolated word correction problem. However, errors of speech recognition tend to be continuous word errors which should be better classified into syntactic and semantic level errors, because the recognizer only produces word sequences existing in a lexicon. So, this section presents a more syntax and semantic-oriented approach to correct erroneous outputs of a speech recognizer using a domain knowledge which provides syntactic and semantic information. We focus on continuous word error detection and correction, using syntactic and semantic knowledge, and pipeline this high-level error correction method with the syllable-based channel model.

4.1 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) (Jung et al., 2003) and a TREC QA system for the purpose of matching the user query with the appropriate answer types at syntax/semantic level (Kim et al., 2001; Lee et al., 2001). 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.’[3] Semantic types con-

[2] We omitted detail character-level match lines to simplify. The whole word match is depicted in bold lines, while no-line means character-level match errors.

[3] Part-of-speech tag denoting a proper noun which is used in Penn TreeBank (Marcus et al., 1994).

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

Table 1: Example of a template abstracted by LSP

sist 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 a company name like 'IBM,' and pre-define 83 semantic category values, such as '@location' for location names like 'New York' (Jung et al., 2003). Figure 4 shows an example of predefined common semantic category values which will be used in an ontology dictionary.

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 a 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', 'query', 'asking', and 'answer.'

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 for common semantic classes reflects the pure general knowledge of the world; hence it performs a supplementary role to extract semantic information. The domain dictionary provides the specific vocabulary which is used in semantic representation tasks of a user query and the template database.

4.2 Construction of a Domain Knowledge

For semantic-oriented error correction, we constructed a domain knowledge, which consists of a domain dictionary, an ontology dictionary, and template queries that are similar to question types in a QA system (Lee et al., 2001). Query sentences are semantically abstracted by LSP's and are automatically collected for the template database.

Because Fujii et al. (Fujii et al., 2002B) have shown the importance of the language model which well describes the domain knowledge, we reflect the domain information with a template database: database of template 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 an ontology dictionary. Assuming that some speech statements for a specific target domain are predefined, a record of the template database is composed of a fixed number of LSP elements, such as POS tags, semantic tags, and domain-specific semantic classes. Table 1 shows an example of template abstracted by LSP conversion in a predefined domain of "on-line education."

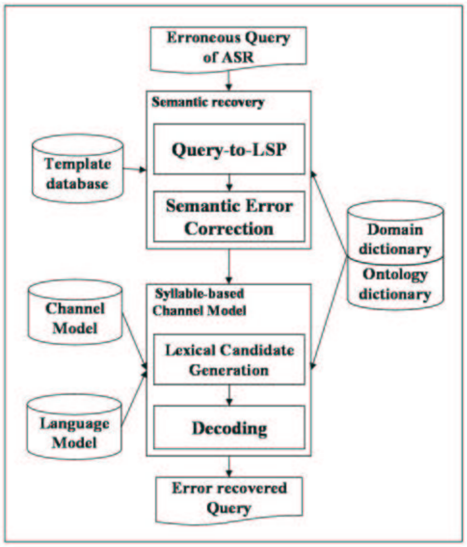


Figure 5: Process of Semantic-oriented Error Correction

Query-to-LSP translation transforms a given query into a 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 words in a query sentence are converted into the LSP through several steps. First, a morphological analysis is performed, which segments a sentence of words into morphemes, and adds POS tags to the morphemes (Lee et al., 2002). NE recognition discovers all the possible semantic types for each word by consulting a domain dictionary and an ontology dictionary. NE tagging selects a semantic type for each word so that a sentence can be mapped into a suitable LSP sequence by searching several types in the semantic dictionaries (An et al., 2003).

4.3 Semantic-oriented Error 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 5 is a schematic diagram of the post error correction process.

The overall process is divided into two stages: a syntactic/semantic recovery and a lexical recovery stage. In the semantic error detection stage, a recognized query is converted into the corresponding LSP. The converted LSP may be ill-formed depending on the errors in the recognized query. Semantic error correction is performed by replacing these syntactic and/or semantic errors using a semantic confusion table. We used a pre-collected template database to recover the semantic level errors, and the technique for searching most similar templates are based on a minimum edit distance dynamic programming search, which has been used as a similarity search in many areas such as spelling correction, OCR post correction, and DNA sequence analysis (Wagner and Fischer, 1974). The semantic confusion table provides the matching cost, which can be semantic similarity, 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 to transform one word into the other. We compute the minimum edit distances between the erroneous LSP's and the template LSP's in the template database using the similarity cost functions at the semantic level, and select, as the final template query, the one which has the minimum distance among them. At this stage, replaced LSP elements can provide some clues of the recognition errors and the original query's meaning to the next lexical recovery stage. Moreover, candidate error boundary can also be detected by this procedure.

After this procedure, lexical recovery is performed in the next stage. Recovered semantic tags and the erroneous queries produced by ASR are the clues of lexical recovery. Erroneous query and recovered template query are aligned by dynamic programming again, after which some lexical candidates are generated by our improved syllable-based channel model. Figure 6 [4] shows an example of semantic error correction process using the same data in TRAIN-95 (Allen et al., 1996).

5 Experiments

5.1 Experimental Setup

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 by 1 male speaker in a real application. We also 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). For our semantic-oriented error correction, we constructed a domain knowledge for our target domain. We constructed 3,195 entries of domain dictionary, 13,154 entries of ontology dictionary, and 436 semantic templates generated automatically using domain dictionary and ontology dictionary.

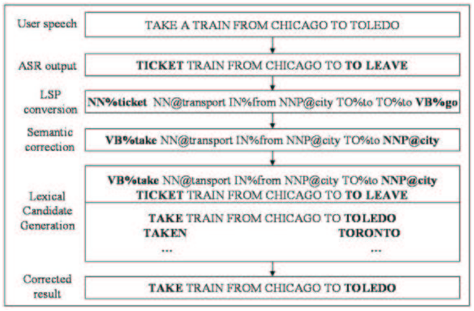

User speech
TAKE A TRAIN FROM CHICAGO TO TOLEDO
ASR output
TICKET TRAIN FROM CHICAGO TO TO LEAVE
LSP conversion
NN%ticket NN@transport IN%from NNP@city TO%to TO%to VB%go
Semantic correction
VB%take NN@transport IN%from NNP@city TO%to NNP@city
Lexical Candidate Generation
VB%take NN@tansport IN%from NNP@city TO%to NNP@city
TICKET TRAIN FROM CHICAGO TO TO LEAVE
TAKE TRAIN FROM CHICAGO TO TOLEDO
TAKEN
TORONTO
...
...
Corrected result
TAKE TRAIN FROM CHICAGO TO TOLEDO

Figure 6: Example of Semantic-oriented Error Correction

We implemented both word-based and syllable-based model for comparison, and combined the system of syllable-based lexical correction with the LSP-based semantic error correction. For experiments, we use trigrams language model generated by SRILM toolkit (Stolcke, 2002), and a training program for channel model made by ourselves. And, we divided the 462 queries into 6 different sets, and evaluated the results of 6-fold cross validation for each model.

5.2 Results

To measure error correction performance, we use word error rate (WER) and term error rate (TER):

$$WER = \frac{|S_w| + |I_w| + |D_w|}{|W_{truth}|} \quad (10)$$

$$TER = \frac{|S_t| + |I_t| + |D_t|}{|T_{truth}|} \quad (11)$$

$|W_{truth}|$ is the number of original words, and $|T_{truth}|$ is the number of query term (or keyword) in original words, that is, an error rate of content words directly related to the performance of IR and QA system (Fujii et al., 2002A).

Table 2, 3 present the experiments results of WER of baseline ASR, word-based channel model, our syllable-based channel model and combined syllable-based channel model with the LSP semantic correction model. The performances of baseline systems were about 79% $\sim$ 81% on the utterances in in-vehicle telematics IR domain. This result shows that the semantic error correction of

[4] In corrected sentence, note that word 'A' is not recovered because this word is meaningless functional word.

Test set	1	2	3	4	5	6	AVG.
Baseline	18.1%	21.6%	19.4%	22.8%	19.9%	19.2%	20.17%
Word-based	12.8%	20.3%	15.5%	17.5%	16.7%	17.7%	16.75%
Syllable-based	10.6%	16.0%	11.5%	16.1%	14.0%	10.6%	13.13%
Syllable + LSP	**9.7%**	**14.9%**	**10.4%**	**15.3%**	**13.0%**	**10.7%**	**12.33%**

Table 2: Result of LG-Elite Recognizer

Test set	1	2	3	4	5	6	AVG.
Baseline	20.5%	18.8%	19.8%	17.0%	16.9%	17.8%	18.47%
Word-based	19.9%	14.8%	18.4%	16.2%	15.3%	15.1%	16.75%
Syllable-based	16.7%	13.8%	17.0%	13.3%	12.7%	12.2%	14.28%
Syllable + LSP	**15.3%**	**13.4%**	**15.8%**	**12.9%**	**11.3%**	**11.8%**	**13.42%**

Table 3: Result of ByVoice

speech recognition result is a viable approach to improve the performance.

Using both baseline ASR systems, we achieved 39% and 27% of error reduction rate. In comparison with the previous word-based model, our new approaches have more accurate error correction performance in this domain. Table 4 shows the result of the experiments for TER. The result of TER shows that baseline ASR systems alone are not appropriate to process the user's queries in speech-driven IR, QA or dialog understanding system. However, with a post error correction, the error reduction rate of TER is much higher than that of WER. And we achieved better performance than word-based model. With this result, our methods are considered to be more appropriate in speech-driven IR and QA applications. Compared with the word-based noisy channel model that has been the best approach in the error correction so far, our semantic-oriented error correction suggests alternative more successful methods for speech recognition error correction.

	Baseline	Word-based	Syllable-based	**Syllable + LSP**
LG-Elite	56.4 %	31.5%	30.1%	**26.7%**
ByVoice	64.1%	34.1%	32.8%	**27.6%**

Table 4: Result of Term Error Rate

6 Conclusion and Future Works

We proposed an improved syllable-based noisy channel model and combined higher level linguistic knowledge for semantic-oriented approach in a speech recognition error correction, which shows a superior performance in domain-specific IR applications.

The previous works only focused on inter-word level error correction, commonly depending on a large amount of training corpus for the error correction model and the language model. So, previous approaches require enormous results of ASR and are dependent on specific speakers and environments. On the other hand, our method takes in far smaller training corpus, and it is possible to implement the method easily and in a short time to obtain the better error correction rate because it utilizes the semantic information of the application domain.

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

And, 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 that can only deal with the morphological variants.

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.

Future work should include the end-performance experiments with IR or QA application for our error correction model.

7 Acknowledgements

This work was partly supported by Jungki KeoJeom Project (MOCIE, ITEP), and by 21C Frontier Project

(MOST).

References

James F. Allen, Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski. 1996. A Robust System for Natural Spoken Dialogue. *In Proceedings of the 34th Annual Meeting of the ACL*

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

J. Barnett, S. Anderson, J. Broglio, M. Singh, R. Hudson, and S.W. Kuo. 1997. Experiments in spoken queries for documents retrieval. *In Proceedings of Eurospeech*, (3):1323-1326.

Eric Brill and Robert C. Moore. 2000. An Improved Error Model for Noisy Channel Spelling Correction. *ACL2000*, 286-293.

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. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79-85

Ciprian Chelba. 1997. A Structured Language Model. *In Proceedings of the Thirty-Fifth Annual Meeting of the ACL and Eighth Conference of the European Chapter of the ACL*, 498-503.

F. Crestani. 2000. Word recognition errors and relevance feedback in spoken query processing *In Proceedings of the 2000 Flexible Query Answering Systems Conference*, 267-281.

Atsushi Fujii, Katunobu Itou, and Tetsuya Ishikawa. 2002A. Speech-driven Text Retrieval: Using Target IR Collections for Statistical Language Model Adaptation in Speech Recognition. *Anni R. Coden and Eric W. Brown and Savitha Srinivasan (Eds.) Information Retrieval Techniques for Speech Application (LNCS 2273)*, 94-104.

Atsushi Fujii, Katunobu Itou, and Tetsuya Ishikawa. 2002B. A method for open-vocabulary speech-driven text retrieval. *In Proceedings of the 2002 conference on Empirical Methods in Natural Language Processing*, 188-195.

Sanda Harabagiu, Dan Moldovan, and Joe Picone. 2002. Open-Domain Voice-Activated Question Answering. *COLING2002*, (1):321-327, Taipei.

Hanmin Jung, Gary Geunbae Lee, Wonseug Choi, KyungKoo Min, and Jungyun Seo. 2003. Multilingual question answering with high portability on relational databases. *IEICE transactions on information and systems*, E-86D(2):306-315.

Satoshi Kaki, Eiichiro Sumita, and Hitoshi Iida. 1998. A Method for Correcting Speech Recognition Using the Statistical features of Character Co-occurrence. *COLING-ACL'98*, 653-657.

Haksoo Kim, Kyungsun Kim, Gary Geunbae Lee, and Jungyun Seo. 2001. 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*

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

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. 2001. 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.

Gary Geunbae Lee, Jeongwon Cha, and Jong-Hyeok Lee. 2002. Syllable pattern-based unknown morpheme segmentation and estimation for hybrid part-of-speech tagging of Korean. *Computational Linguistics*, 28(1):53-70.

Mitchell P. Marcus and Beatrice Santorini and Mary Ann Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313-330.

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

Andreas Stolcke 2002. SRILM - An Extensible Language Modeling Toolkit. *In Proceedings of Intl. Conf. on Spoken Language Processing*, (2):901-904, Denver, Co. (http://www.speech.sri.com/projects/srilm/)

Robert A. Wagner and Michae J. Fischer. 1974. The String-to-String Correction Problem. *Journal of the ACM*, 21(1):168-173.

I. Witten and T. Bell. 1991. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *In IEEE Transactions on Information Theory*, 37(4).