Semantic-Oriented Error Correction for Voice-Activated Information Retrieval System

Yongwook Yoon(POSTECH), Byeongchang Kim(UIDUK Univ.), Gary Geunbae Lee(POSTECH)

<Contents>

<Abstract>

Semantic-Oriented Error Correction for Voice-Activated Information Retrieval System

Yongwook Yoon, Byeongchang Kim, Gary Geunbae Lee

Voice input is often required in many new application environments, but the low rate of speech recognition makes it difficult to extend its application. Previous approaches were to raise the accuracy of the recognition by post-processing of the recognition results, which were all lexical-oriented. We suggest a new semantic-oriented approach in speech recognition error correction. Through experiments using a speech-driven in-vehicle telematics information application, we show the excellent performance of our approach and some advantages it has as a semantic-oriented approach over a pure lexical-oriented approach.

* Key word: Speech recognition, Error correction, Semantics

1. Introduction

A new application environment such as mobile information retrieval requires 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 raise the accuracy of information retrieval, appropriate post processing is required such as error correction. The average recognition accuracy of a representative continuous ASR system is not more than 90% in the word-based and at most 70% in the sentence-based (Fujii et al., 2002).

Error correction for speech recognition is different from the correction ofwritten texts or of optical character recognition (OCR). A statistical method using some linguistic knowledge was before applied to the correction of OCR errors (G. Lee et al., 1997). However, the characteristics of speech recognition errors are far from those of errors in written text recognition. So, the methodologies applied to the correction of written text recognition often fail to show satisfactory performance in correcting speech recognition errors.

There have been several studies on the correction of speech recognition errors. Because, in most cases, the speech recognition is not a target task itself but an intermediate process of an entire application system, the error correction task is closely related to the specific application where it is used, while a few cases deal with the correction task in its own. Also, it seems that most works take slightly different approaches in dealing with error correction, according to the characteristics of the application in which they work. For example, one approach (Ringger and Allen, 1996) suggests a method independent of the application type to which it is applied. Another approach (Kaki et al., 1998) produces a correction which can only be used in a specific application, and the other work (Hanmin Jung et al., 2001) does not even produce any corrected lexical words but produces some information about the corrected words which helps in making better performance of the application system.

Most previous researches 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 (Ringger and Allen, 1996 Kaki et al., 1998). The performance of such systems depends on the size and quality of the corpus or the database of error strings they have collected.

They use the probability of a word to be mistakenly recognized, 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 somewhat successful results, but they still have some drawbacks. The error patterns constructed are available but are not abundant, because it costs much expense to collect themthere are so many cases where they fail to recover the original strings from 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 the drawbacks of the previous fragile lexical based approaches. In our approach, in addition to the lexical information, we use some semantic information that the words in speech transcription have. We obtain semantic information from some knowledge base such as general thesauri and a special dictionary which we construct by ourselves to contain domain knowledge specific to the target application.

The semantic-oriented approach has some strength to the lexical based one, since it is less sensitiveto each error pattern. Also, it has more broad coverage of an error pattern, since several similar error strings common, in a sense,can be reduced to one semantic error pattern, which enables us to improve the probability of recovering from erroneous recognition results.

2. Related Works

We will show some examples of speech recognition errors (See Ringger and Allen, 1996) before mentioning the details of the different approaches of error correction. First, it is the case that one original word is replaced with a different word due to recognition error.

U : Right send the train from Montreal
R : Rate send that train from Montreal

In the above example, "U" denotes the original sentence a user speaks, and "R" denotes the text a speech recognizer produces as a result of the recognition. Occasionally, there happens the case where more than one word is replaced with a word of the original sentence and vice versa (M-to-N mapping).

U : Take a train from Chicago to Toledo
R : *Ticket train from Chicago to to leave*

We would specially call these cases M-to-N replacement errors. Next, one or more words may be inserted to a location of the sentence, or one or more words are deleted from the original sentence.

U : great okay now we could go from
R : I'm great okay week it go from

Ringger and Allen (1996) suggested the noisy channel model for error correction. They tried to construct a general post-processor that can correct errors generated from any speech recognizer. The model consists of two parts: a channel model, which accounts for errors made by the SR (speech recognizer), and the language model(See Jelinek, 1990), which accounts for the likelihood of a sequence of words being uttered in the first place.

They trained the channel model and the language modelboth 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. Their model is elegant and their system showed a moderate error correction rate, but they failed to cope with M-to-N replacement errors successfully.

Kaki et al. (1998) suggested a straightforward and intuitive method to robustly handle many kinds of recognition errors. They collected as 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 including an M-to-N replacement 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 Fujii et al. (2002), an indirect error correction is introduced. They suggest a new type of information retrieval system tightly integrated with a speech input interface. In their system, the collection documents provide an adaptation of the language model of the SR, which results in a drop of the word error rate. Through the same framework, they also tried to solve the out-of- vocabulary word problem, which is another

important issue in the SR (speech recognition) and IR (information retrieval) integration.

The previous works commonly require a large amount of training corpus on the error model and the language model. On the other hand, our method takes far less training corpus, and it is possible to implement easily and in short time to obtain the same or better error correction rate because it utilizes thesemantic information of the application domain.

3. Semantic Oriented Approach

3.1. Lexico Semantic-Pattern

A lexico-semantic pattern (LSP) is the structure where linguistic entries and semantic types can be used in combination to abstract certain sequences of the words in a text. It has been used in the area of NLIDB (Hanmin Jung et al., 2001) and TREC QA system (G. Lee et al., 2001 H. Kim et al, 2001) for the purpose of matching the user querywith the appropriate answer types the system wants.

In LSP, linguistic entries consist of words, phrases and part-of-speech tags, such as "Shop", "Gas Station", and "ncp" ("ncp" means a noun in our part-of-speech tag symbols.). Semantic types consist of common semantic tags (semantic categories) and user-defined specific semantic classes. The common semantic tags again include attribute-values in databases, such as '@corp' for "SAMSUNG" and "LG" (Korean companies) and 83 semantic category values (See Table 1), such as '@name' for "Sinchon We use the Yale Romanization for Korean letters." (a Korean place name) and "Ka@nam" (also, a Korean place name).

<Table 1> 83 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	@exam	@organization	@status	@unit_time
@country	@hobby	@method	@subject_area	@unit_volume
@date	@law	@address	@substance	@unit_weight
@direction	@level	@appliance	@team	@unit_age
@disease	@living_part	@art	@transport	
@drug		@computer	@weekday	
		@course	@picture	
		@deed	@river	
			@room	
			@sex	

The user-defined semantic classes include special attribute names in databases, such as '%together' for "Kachi" (together) and "Hamkkey" (also, together) and semantic category names, such as '%person' for "fireman" and "teacher", for which the user wants specific meaning in the application domain.

The words in a query sentence are converted into the LSP patterns through several steps. First, a morphological analysis(Jeongwon Cha et al., 1998) is performed, which segments a sentence of words into morphemes and adds a tag of POS(part-of-speech) label to the word. Next, each word of the sentence is converted into a suitable semantic symbol by searching up several types of semantic dictionaries.

We constructed the two semantic category dictionaries which are the domain dictionary and the query dictionary(For the details, see Section 3.2). The semantic category dictionary consists of two components: semantic tags and user-defined semantic classes. Left-hand side is for keywords and right-hand side for semantic information. Multi-words are described at the end of the entries according to their first keywords, for example, "LG" leads to "LG" and "LG Oil". All attribute-values are automatically extracted from databases and are added into the dictionary with the form of left-hand side (values) and right-hand side (their attributes).

[Semantic tags in category dictionary]

SK (@corp)

LG (@corp| '|' in parenthesis means "OR" separator.@name)
| '|' is the separator for multi-word entries.LG_Oil_@corp:
The word following ':' is a normalized word used in our pattern databases.LG
LG Cwuyuso (@name)
Sinchon (@subway)
|Sinchon_yek_@subway:Sinchoyek

[User-defined semantic classes in category dictionary]
Kakyek (%price)
|Kakyeki_nac_un_%price-asc
Iyo@i (%action)
Ceyil (%most)
|Ceyil_ssa_%most-cheap

The structure of semantic tags is a flat form. In a lexico-semantic pattern, each semantic tag follows a '@' symbol. For example, a semantic tag '@location' includes the words, such as "Tapa@" (coffeeshop) and "Pye@wen" (hospital). User-defined semantic classes are the tags for syntactically or semantically similar lexical groups. We use the classes to abstract out several synonyms into a single concept. For example, a user-defined semantic class '%each' represents the words, such as "Kak", "Kakphwummok", "Kaybyel", and "Byel"(All, each or every in English).

3.2. Source LSP Database for Specific Domain

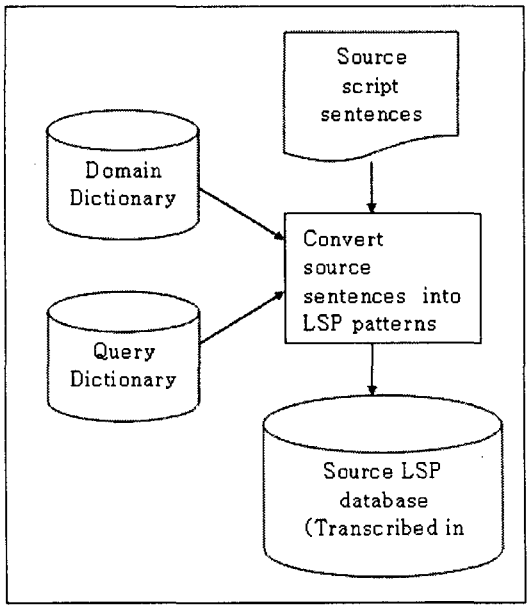
As (Fujii et al., 2002) has showed the importance of the language model which well describes the domain knowledge, we reflect the domain information with special semantic databases: LSP database of the source statements. The source LSP database is automatically acquired by the LSP translation using two dictionaries: domain dictionary and query dictionary (Figure 1).

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 query 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 sourcedatabase of LSP transcription is for the actual error detection and correction task after speech recognition.



<Figure 1> Construction of LSP database

Assuming that some speech statements for a specific target domain is predefined, a record of the source database is made up of a fixed number of LSP elements, such as POS tags, semantic tags, and user-defined semantic classes. Figure 2 shows examples of the source LSP database which plays an error correction model against input speech transcripts.

Source sentence: Seychaca@kwa phyeunycemi issnun

Transcribed LSP patterns:

%wash|%and|%serv|jcs|@yokwupa|jxc	*
*|%and|%serv|jcs|@yokwupa|jxc	%wash
%wash|*|serv|jcs|@yokwupa|jxc	%and
%wash|%and|*|jcs|@yokwupa|jxc	%serv
%wash|%and|%serv|*|@yokwupa|jxc	jcs

<Figure 2> Contents of the source LSP database

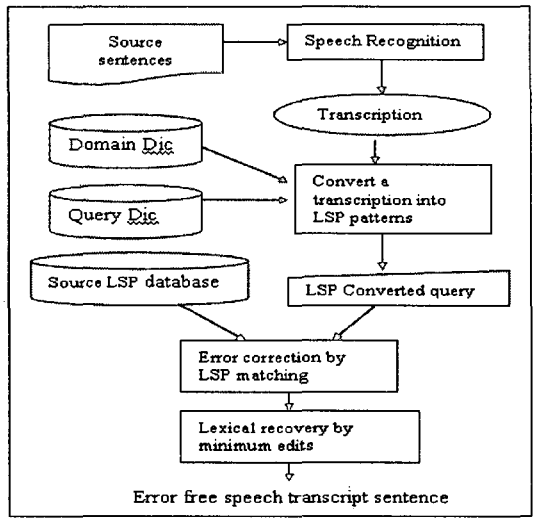
On the average, a sentence of the source script is converted into from 15 to 20 LSP elements in the source LSP database. In the database, one record is composed of 6 LSP elements (window size 6) which are separated by a '|', plus one LSP element separated by a tab. In the left column, seven similar patterns come out repeatedly as the only one element among 6 is replaced with a '*' except for the first pattern. As shown later, these similar lines of LSP patterns are compared to a user spoken query in a consecutive manner in the error-correction process.

3.3 Error Correction based on LSP Pattern Matching

Now we will show the working mechanism of error correction of a speech recognition result using source LSP database.

Suppose we obtain the speech recognition result containing an error as this: "Seychaca@kwa **phyeuycem**i issnun kuncheuy cwuyusolul chacacwe" ("Please tell me where is the gas-station, which has a car wash and a shop together"). Here, "phyenuycem" (shop) is misrecognized as "phyeuycem".

As described in section 3.1, the transcript sentence is first converted into the LSP pattern. Its starting point would be read as "%wash|%and|**uknc**|jcs|@yokwupa|jxc|...". The third LSP element 'uknc' denotes a POS tag with a meaning of "unknown word", caused by the recognition error, and the system failed to find out thecorresponding LSP in the source LSP database due to the recognition error "phyeuycem".



<Figure 3> Error correction by LSP matching

The converted LSP pattern is divided into a substring ofsix consecutive LSP elements, and then compared to each pattern in the source LSP database, where the 'uknc' symbol in the query pattern is replaced with a '*' symbol. It is a type of sliding window comparison method.

In this example, we would expect to find out the pattern "%wash|%and|*|jcs|@yokwupa|jxc" in the database as shown at 4th line in Figure 2. In the right column of this line, we can see the element '%serv' which is a user-defined semantic class for "phyenuycem", one of the candidates of error correction. After this procedure, many lexical correction candidates under the same semantic class are produced, and we select one as the final correction word using minimum edit distance.

The "Minimum Edit Distance" between two words is defined as the minimum num ber of deletions, insertions, and substitutions required transforming one word into the ot her. We compute the minimum edit distances between the error word and the lexical c orrection candidates by the algorithm from (Wagner and Fischer, 1974), and select as t he final correction word the one which is of the shortest distance among them.

In Figure 4, we show an entire scenario where a user query is transcribed by the SR, and the erroneous word is detected and corrected into the correct original one.

Input Query: "Seychaca@kwa **phyeuycem**i hamkey issnum cwuyuso cwu@eyse kakyeki kaca@ celyem han kos?"

LSP Matching:

# [0] [0->0]	Seychaca@	ncn (1 0)		%wash
# [1] [1->1]	kwa	j (0 0)	%and	
# [2] [2->2]	**phyeuycem**	uknc (1 0)	**uknc**	
# [3] [3->3]	i	j (0 0)	jcs	
# [4] [4->4]	hamkey	ma (1 0)	%together	
# [5] [5->5]	iss	pa (1 0)	@yokwupa	
# [6] [6->6]	nun	ef (0 0)	jxc	
# [7] [7->7]	cwuyuso	unoun (1 0)	%name	
# [8] [8->8]	cwu@	unoun (1 0)	nbn	
# [9] [9->9]	eyse	j (0 0)	%from	
# [10] [10->10]	kakyek	ncn (1 0)	%price	

```
# [11] [11->11]i          j (0 0)          jcs
# [12] [12->14]kaca@  celyem ha     ncp (1 0)          %most-cheap
# [13] [0->0]                      (1 0)
# [14] [0->0]             (0 0)
# [15] [15->15]n          ef (0 0)          ef
# [16] [16->16] kos                ncn (1 0)          %location

# Speech Recognition Error: %wash|%and|uknc|jcs|%together|@yokwupa
# Speech Recognition Hypothesis: %wash|%and|*|jcs|%together|@yokwupa      ?
%shop|ncn

# Speech Recognition Error: %and|uknc|jcs|%together|@yokwupa|jxc
# Speech Recognition Hypothesis: %and|*|jcs|%together|@yokwupa|jxc      ?
%serv|%shop|%wash|ncn

# Speech Recognition Error: uknc|jcs|%together|@yokwupa|jxc|%name
# Speech Recognition Hypothesis: *|jcs|%together|@yokwupa|jxc|%name      ?
%shop|%wash|ncn

The most probable error detected: pheyuycem
Correction candidates stat. (frequency): %shop (3), ncn (3), %wash (2), %serv (1)
LSP significance level: %shop > ncn                    semantic class is higher in
significance level than POS

# Speech Recognition Correction: uknc ? %shop  semantic level correction

# %shop: siksa, umsik, phyenuycem                    the entries in the semantic
category dictionary for '%shop'
# The final correction word -> phyenuycem                    by the minimum edit
distance from ""phyeuycem"
```

<Figure 4> The entire error correction scenario for a user's spoken query

4. Evaluation

4.1 Experiment Environment

We performed experiments on the domain of In-Vehicle Telematics information retrieval related to gas-station services. The speech transcripts used in the experiments are composed of 1,011 statements. We used a commercial speech recognizer, ByVoice Refer to http://www.voicetech.co.kr, trained with an acoustic and language model for Korean.

ByVoice is the leading edge in Korean continuous speech recognizer market, and it requires the speaker adaptation process about 30 minutes. If additional learning process for domain dependent language model in which the speech is applied is performed, the recognition rate is considerably improved. So we performed the language model adaptation using our domain transcript, the above 1011 statements.

Among the 1011 statements, we selected 540 sentences to include the various error types, including typical insertion, deletion, and replacement errors. Table 2 shows the speech recognition results without any error correction. It would be the baseline performance of our system.

<Table 2> Speech recognition performance (baseline)

Recognition Level	# of input	# of incorrect	Accuracy Rate
Sentence	540	224	58.5 %
Word	4243	341	92.0 %

4.2 Experiment Results

As represented in Table 2, among 540 transcript sentences, the 224 sentences have contained the recognition errors. These 224 statements were processed through our error correction system. The word level correction results classified by error types are shown in Table 3.

<Table 3> Error correction rates (word level)

Error type	Ins	Del	Repl	Total
Erroneous word	15	9	95	119
Lexically Corrected word	8	4	55	67
Precision_1	53.3	44.4	57.9	56.3
Semantically Corrected word	15	9	71	95
Precision_2	100	100	74.7	79.8

We define precision as the ratio of the number of corrected sentences to the number of the sentences detected as containing recognition errors. The Precision_1 denotes the precisionvalue when the lexically corrected cases are regarded.

With the lexically corrected results as the input of the NLIDB application (Hanmin Jung et al., 2001), we examined whether the NLIDB system yields a correct retrieval output. According to these retrieval results, we determined another precision value, Precision_2, which can be regarded as semantically correct results.The total Precision_2 79.8% is higher than Precision_1 56.3% by 42%, which indicates that our semantic oriented approach shows a good performance when connected with an application system which handles linguistic semantic information.

The overall recognition accuracy after the error correction is 93.5% in word level. Compared to the baseline performance 92.0%, it represents a considerable improvement.

Thus, our semantic-oriented approach has the following advantages: First, it is fast and easy, and renders computationally simple implementation. The background knowledge dictionaries are constructed only once, except for the domain dictionary which is dependent ona specific application domain, but is very small compared to the general semantic dictionary (the query dictionary). And another required database, the source LSP database, can be constructed automatically in a short time using our semantic dictionaries. It also means that our approach enables us to make an easy adaptation to the change in the application domain.

Second, because the LSP scheme transforms pure lexical entries into semantic categories, the size of the error pattern database can be reduced remarkably, and it also increases the coverage and robustness of 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 any speech-driven natural language information retrieval system.

5. Conclusion

We proposed a semantic-oriented approach in the speech recognition error correction. Our approach shows a better performance comparedto the previous methods of pure lexical-oriented approaches. In addition, our method has some advantages, such as easy and quick implementation, the flexibility in the change of application domain, and an excellent combination with applications of speech-driven information retrieval.

However, there are some unsolved problems: M-to-N replacement errors, massive number of insertion or deletion of words in a single sentence, and out-of-vocabulary word problem which hinders the improvement in the retrieval performance. All these problems require for future active works for the semantic oriented speech error correction.

References

Allen, James F. & Eric K. Ringger (1996), A Fertility Model for Post Correction of Continuous Speech Recognition, *ICSLP'96*, pp.897~900.

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

Atsushi Fujii, Katunobu Itou, Tetsuya Ishikawa. 2002. A method for open-vacabulary speech-driven text retrieval, *Proceedingof the 2002 conference on Empirical Methods in Natural Language Processing,* pp.188~195.

Cha, Jeongwon, Geunbae Lee and Jong-Hyeok Lee (1998), Generalized unknown morphemeguessing for hybrid POS tagging of Korean, *Proceedings of SIXTH WORKSHOP ON VERY LARGE CORPORA in Coling-Acl 98*, Montreal, pp.85~93.

F. Jelinek (1990), Self-Organized Language Modeling for Speech Recognition, *Readings in Speech Recognition,* Morgan Kaufmann Publishers, Inc., San Mateo, CA, pp.450~506.

Fischer, J. Michael & Robert A. Wagner (1974), Thestring-to-string correction problem, *Journal of the ACM 21(1)*, pp.168~173.

Geunbae Lee, Jong-Hyeok Lee, Jinhee Yoo (1997), Multi-level post-processing for Korean

character recognition using morphological analysis and linguistic evaluation, *Pattern recognition 30(8)*, pp.1347~1360.

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, *Proceedings of the 10th Text Retrieval Conference (TREC-10)*, Washington D.C.

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

Hanmin Jung, Gary Geunbae Lee, Wonseug Choi, KyungKoo Min and Jungyun Seo (2001), Multi-lingual question answering with high portability on relational databases, *IEICE transactions on information and systems* (in press).

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*, pp.653~657.

접수일자: 2002년 11월 13일
게재결정: 2002년 12월 12일

▶ Yongwook Yoon
address : San 31, Hyoja-Dong, Pohang, 790-784, Korea (South)
affiliation : Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH)
E-mail : ywyoom@postech.ac.kr
Tel : 054) 279-2254
Fax : 054) 279-2299

▶ Byeongchang Kim
address : Gangdong, Gyeongju, 780-713, South Korea
affiliation : Division of Computer and Multimedia Engineering, UIDUK University
E-mail : bckim@uiduk.ac.kr
Homepage : http://www.uiduk.ac.kr/~bckim
Tel : 054) 760-1657

▶ Gary Geunbae Lee
address : San 31, Hyoja-Dong, Pohang, 790-784, Korea (South)
affiliation : Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH)
E-mail : gblee@postech.ac.kr
Homepage : http://nlp.postech.ac.kr/~gblee/
Tel : 054) 279-2254
Fax : 054) 279-2299