

Integrated speech and morphological processing in a connectionist continuous speech understanding for Korean

Geunbae Lee

Jong-Hyeok Lee

Department of Computer Science & Engineering

Pohang University of Science & Technology

San 31, Hoja-Dong, Pohang, 790-784, Korea

Phone: +82-562-279-2254, Fax: +82-562-279-2299

gblee@vision.postech.ac.kr, jhlee@vision.postech.ac.kr

Abstract

A new tightly coupled speech and natural language integration model is presented for a TDNN-based continuous possibly large vocabulary speech recognition system for Korean. Unlike popular n-best techniques developed for integrating mainly HMM-based speech recognition and natural language processing in a *word level*, which is obviously inadequate for morphologically complex agglutinative languages, our model constructs a spoken language system based on a *morpheme-level* speech and language integration. With this integration scheme, the spoken Korean processing engine (SKOPE) is designed and implemented using a TDNN-based diphone recognition module integrated with a Viterbi-based lexical decoding and symbolic phonological/morphological co-analysis. Our experiment results show that the speaker-dependent continuous *eojeol* (Korean word) recognition and integrated morphological analysis can be achieved with over 80.6% success rate directly from speech inputs for the middle-level vocabularies.

Keywords: speech and natural language integration, spoken language processing, morphological analysis, phonological modeling, Viterbi search, time-delayed neural networks

1 Introduction

A spoken natural language system requires many different levels of knowledge sources including acoustic-phonetic, phonological, morphological, syntactic, semantic and even pragmatic levels. These knowledge sources are grouped and processed by either speech processing models or statistical/symbolic natural language processing models. Since the speech and the natural language communities have conducted almost independent researches, these models were not completely integrated and often biased by neglecting either acoustic-phonetic or high-level linguistic information. Current speech and natural language integration mainly relies on word-level n-best search techniques [1, 2] as shown in figure 1. For HMM-based speech recognition systems, the n-best search techniques have been successfully applied to the integration of speech and natural language processing. However, current implementations of n-best techniques only support the integration at a word level by directly producing the n-best list of candidate sentences, and this type of loose coupling is

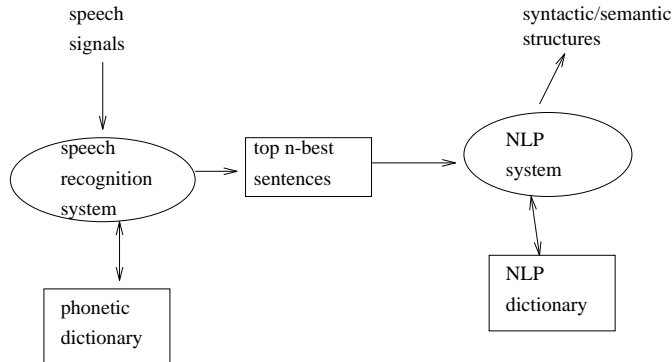


Figure 1: N-best lists: current speech and natural language integration method

only suitable for the integration of existing speech and natural language systems, such as, e.g. [3, 4]. The n-best search is viable only for short sentences since the necessary n grows exponentially with the sentence length. Because the n-best search directly generates word sequences, phonetic and natural language dictionaries must have full word entries, which is obviously inadequate for morphologically complex agglutinative languages such as Korean. The dictionary size will grow very fast for full word entries because new words can be almost freely generated by concatenating the constituent morphemes in these languages (e.g. noun plus noun-endings or verb plus verb-endings).

In this paper, we present a new morphologically conditioned integration architecture of speech and natural language processing for morphologically complex agglutinative languages. The integration is based on a Viterbi-based lexical decoding and symbolic phonological/morphological co-analysis. The Viterbi search [5] is performed on diphone (explained in section 4) sequences generated from a TDNN (time-delay neural network)-based Korean speech recognizer [6], and the search process is tightly integrated with a morphological and phonological constraint checking. We present a new integration architecture, not for popular HMM-based systems, but for recently developed connectionist speech recognition systems. Connectionist speech recognition has several advantages over the classical statistical speech processing [7]. Especially, the TDNN model [8] has been widely used to model the time shift invariance of speech signals. In this regard, we will present a morpheme-level integration method for a TDNN-based continuous speech recognition model for Korean. This paper is organized as follows. Section 2 briefly explains the characteristics of spoken Korean for general readers. Section 3 introduces our speech and natural language integration architecture, and section 4 and section 5 more elaborate the introduced integration architecture. Section 6 shows several experiment results to demonstrate the performance, and section 7 compares our integration scheme with similar related researches. Section 8 draws some conclusions.

2 Features of spoken Korean

This section briefly explains the linguistic characteristics of spoken Korean before describing the integration architecture. In this paper, Yale romanization is used for representing the Korean phonemes. 1) A Korean word, called *eojeol*, consists of more than one morphemes with clear-cut morpheme boundaries (Korean is an agglutinative language). 2) Korean is a postpositional language with many kinds of noun-endings, verb-endings, and prefinal verb-endings. These functional

morphemes determine the noun’s case roles, verb’s tenses, modals, and modification relations between *eojeols*. 3) Korean is a basically SOV language but has relatively free word order compared to the rigid word-order languages, such as English, except for the constraints that the verb must appear in a sentence-final position. However, in Korean, some word-order constraints do exist such that the auxiliary verbs representing modalities must follow the main verb, and the modifiers must be placed before the word (called head) they modify. 4) The unit of pause in speech (which is called *eonjeol*) may be different from that of a written text (an *eojeol*). The spoken morphological analysis must deal with an *eonjeol* (fragment of sentence) since no *eojeol* boundary is provided in the speech. 5) Phonological changes can occur in a morpheme, between morphemes in an *eojeol*, and even between *eojeols* in an *eonjeol*. These changes include consonant and vowel assimilation, dissimilation, insertion, deletion, and contraction, and so on.

3 SKOPE system architecture for morpheme-level integration

The morpheme-level integration technique processes phoneme-like unit (PLU) sequences (speech recognizer’s outputs) using both Viterbi-based lexical decoding (for morpheme) and symbolic phonological/morphological co-analysis, and uses a single unified phonetic-morpheme (UPM) dictionary for both speech and language processing. This morpheme-level integration scheme is able to utilize natural language morphological processing techniques in an early stage of spoken language processing compared with the classical approaches of word-level speech and language integration. The morpheme-level integration also renders a phonological rule modeling possible in the early stage. The phonological/morphological analysis can be performed together using the single UPM dictionary, and the dictionary size becomes stable regardless of the vocabulary size because only the morphemes are encoded and the new words can be processed by using the existing morphemes in the dictionary.

Figure 2 shows the SKOPE architecture, a morpheme-level integration model of speech and natural language processing for Korean. The speech signal is analyzed using the TDNN diphone recognizer. The diphone recognizer is composed of a hierarchy of TDNN networks. The recognized diphone sequences are decoded using the Viterbi search on the trie-structured UPM dictionary to segment out the target morpheme candidates. In the UPM dictionary, each morpheme’s phonetic header is a HMM (hidden markov model) network using the diphone symbols. The Viterbi decoded candidate morphemes are stored in a triangular table to be properly connected during the morphological processing. From the candidate morphemes, the Viterbi-based morphological analyzer produces the morphologically analyzed *eojeols* by handling morphotactics verification and irregular conjugations. The phonological modeling is tightly integrated into the morphological processing through a declarative phonological rule modeling in the UPM dictionary. Outputs of the integrated architecture, that is, analyzed *eojeol* sequences, can be directly fed to the upper level syntax and semantics analysis modules which are described in [9].

4 Diphone-based speech recognition

For large-vocabulary continuous speech recognition, a sub-word level recognition is usually performed. We select a group of diphones for our phoneme-like units (PLUs) because direct phoneme recognition in Korean is very difficult. The 46 Korean phonemes are very similar each other espe-

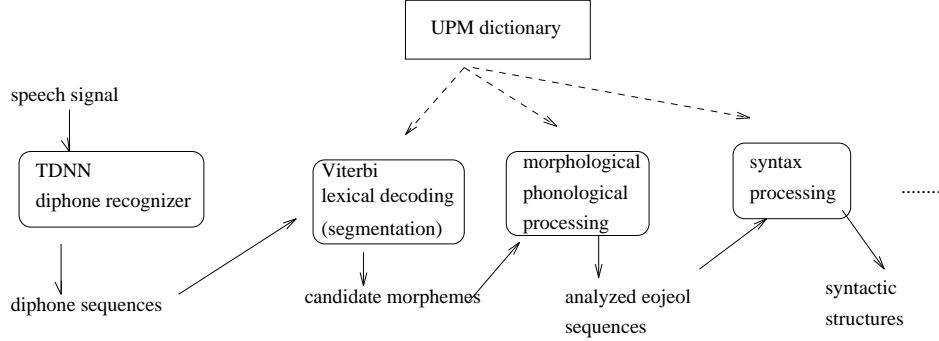


Figure 2: SKOPE speech and language morpheme-level integration architecture. Syntax and more high-level processing steps are not in the scope of this paper.

diphone types	diphone numbers	diphone examples
V	21	a, o, wu, i, u, ye,
C1V	378	ha, sa, ka, la, ma, kha,
VC2	147	an, am, eng, em, wun, in,
C2C1	72	ngs, nn, ngd, ngh,

Figure 3: Korean diphone groups (V: vowel, C1: syllable-first consonant, C2: syllable-final consonant). In C2C1 type, the C2 must be one of the nasals or liquids/glides which are similar to vowels. Yale romanization is used to specify the diphone symbols.

cially in the following cases: 1) the Korean diphthongs are hard to distinguish from the mono-vowels, and 2) the syllable-final consonants are hard to differentiate from the syllable-first consonants. The selected diphone groups (figure 3) have more suitable features for co-articulation modeling than the phonemes and are much fewer in numbers than the popular triphones [10]. We also introduced CC-type (syllable-final consonant, syllable-first consonant) diphones for smooth transition modeling between syllables in Korean. Figure 4 shows the hierarchical structure of a group of TDNNs for diphone recognition, and also shows the architecture of each component TDNN. The whole diphone recognizer consists of total 19 different TDNNs for recognition of the defined Korean diphones. We re-classified the total diphones into 18 different groups according to the vowel characteristics in the diphones. The top-level TDNN (vowel group TDNN) identifies the 18 vowel groups of the diphones using relatively low frequency signal vectors (under 4 KHz). Each 18 different sub-group TDNN recognizes the target diphones using the whole frequency signal vectors. For the training of each TDNN, we manually segmented the digitized speech signals into 200 msec range (which includes roughly left-context phoneme, target diphone, and right context phoneme), and applied 512 order FFTs and 16 step mel-scaling [8] to get the filter-bank coefficients. Each frame size is 10 msec, so 20 (frames) by 16 (mel-scaling factor) values are fed to the TDNNs with the proper output symbols, that is, the vowel group name or the target diphone name. After the training of each TDNN, the diphone recognition is performed by feeding 200 msec signals to the vowel group TDNN and subsequently to the proper sub-group TDNNs according to the extracted vowel group. The 200 msec signals are shifted by 30 msec steps and continuously fed to the networks to process the

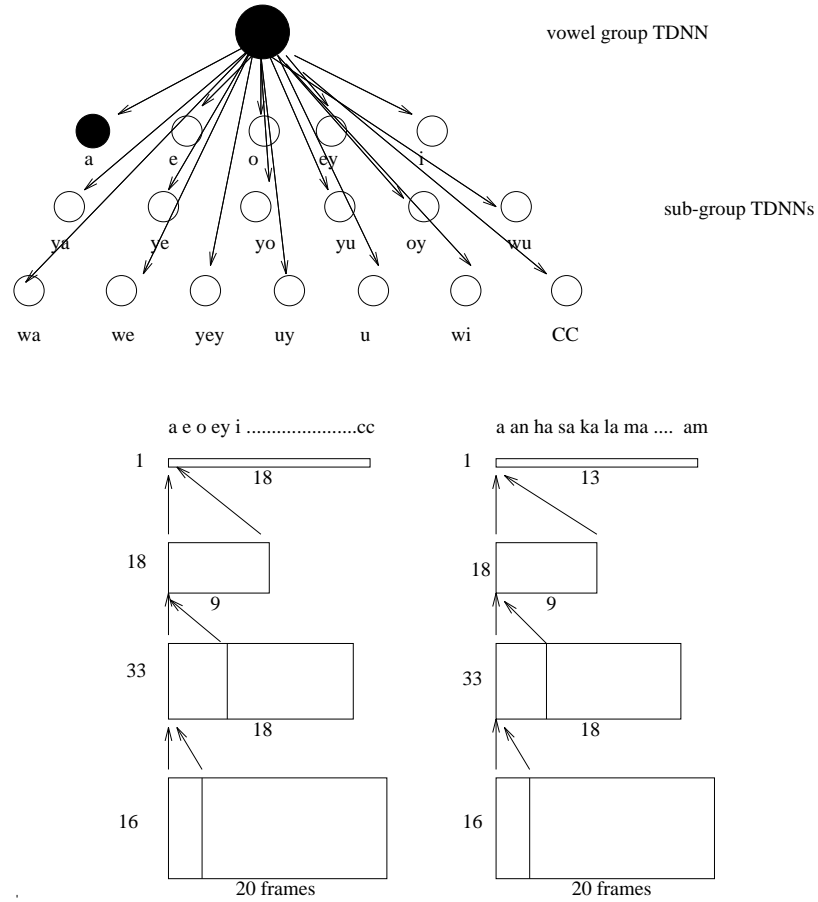


Figure 4: Top: hierarchical organization of the group of TDNNs for entire diphone recognition. Bottom left: TDNN architecture for vowel group identification. Note the cc group contains no vowels. Bottom right: Architecture of the sub-TDNN for /a/ vowel group recognition. The other 17 sub-TDNNs have the same architecture, but different number of output units according to the number of diphones in each of the vowel group.

phonetic transcription header	original morpheme	left morphological connectivity	right morphological connectivity	left phonemic connectivity	right phonemic connectivity
ci-wu	ci-wu	regular verb	regular verb	'c' sound no-change	'wu' sound no-change
l	l	adnominalizing verb-ending	adnominalizing verb-ending	'l' sound no-change	'l' sound no-change
sswu	swu	bound-noun	bound-noun	's' sound change to 'ss'	'wu' sound no-change

Figure 5: The unified phonetic-morpheme (UPM) dictionary for entries *ci-wu* (delete), *l* (adnominalizing verb-ending), and *sswu* (bound-noun).

continuous speech in an *eonjeol* (pause unit of Korean speech). The final outputs are sequence of diphones for each 200 msec range in 30 msec intervals. The hierarchical TDNN structure shortens the training time and provides easily extensible system design. The entire recognition rate critically depends on the vowel group TDNN in this hierarchical structure.

5 Viterbi-based morphological analysis

Unlike conventional morphological analyses for text inputs, our morphological analysis starts with the recognized diphone sequences which contain insertion, deletion, and substitution speech recognition errors. The conventional morphological analysis procedure [11], i.e., morpheme segmentation, morphotactics modeling, and orthographic rule (or phonological rule) modeling, must be augmented and extended to cope with the recognition errors as follows: 1) The conventional morpheme segmentation is extended to deal with the speech recognition errors and between-morpheme phonological changes as well as irregular conjugations during the segmentation, 2) the morphotactics modeling is extended to cope with the complex verb-endings and noun-endings in Korean, and 3) the orthographic rule modeling is combined with the phonological rule modeling to correctly transform the diphone transcriptions (phonetic spelling) into the orthographically spelled morpheme sequences.

The central part of the morphological analysis lies in the dictionary construction. In our UPM (unified phonetic-morpheme) dictionary, each phonetic transcription of single morpheme has a separate dictionary entry. Figure 5 shows the UPM dictionary both for speech and language processing with three different morpheme entries *ci-wu*, *l*, *sswu*. The extended morphological analysis is based on the well-known tabular parsing technique for context-free languages [12] and augmented to handle the Korean phonological rules and speech recognition errors in the diphone sequence inputs. Figure 6 shows the extended table-driven morphological analysis process. The example diphone sequence was obtained from the input speech *ci-wul-sswu* (meaning: can/cannot be removed), and the morphological analysis produces *ci-wu+l+sswu* (remove+ADNOMINAL+BOUND-NOUN), where '+' is the morpheme boundary, and '-' is the syllable boundary. The morpheme segmentation is basically performed using the Viterbi-based lexical decoding to recover the possible errors in the diphone sequences. For Viterbi search, the phonetic transcription headers for each morpheme in the UPM dictionary are converted into diphone transcription headers, and each converted header

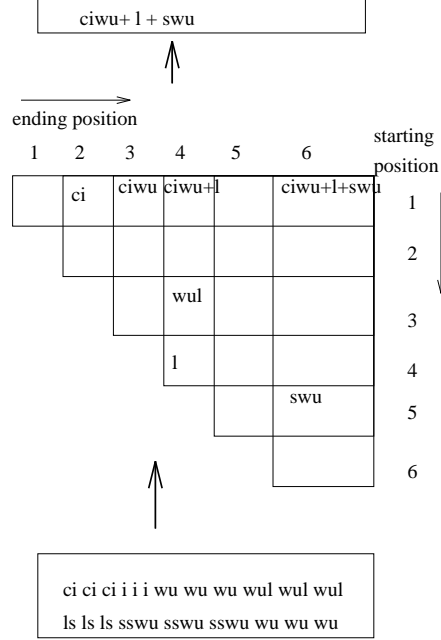


Figure 6: Morphological analysis of diphone sequences. From top: output morpheme sequence in an *eonjeol*, triangular parsing table, and input diphone sequence.

is turned into a simple HMM. The converted HMMs are organized into a trie data structure for efficient search (see figure 7), and form a trie-structured diphone-based HMM index. The HMMs are the simplest ones which have only left-to-right and self transitions. Additional diphone nodes (marked with thick circles) are inserted for smooth inter-morpheme co-articulation modeling. The transition probability in each HMM is defined:

$$a_{ij} = \begin{cases} \alpha & i = j \\ \frac{1-\alpha}{N} & i \neq j \wedge d^t = s_i \wedge d^{t+1} = s_j \\ 0 & \text{otherwise} \end{cases}$$

where a_{ij} is a transition probability from state i to state j , N is the number of all possible transitions from state i . d^t is a diphone observable at time t , and s_i is a diphone at state i . This model assigns self-transition probability α and left-to-right transition probability $\frac{1-\alpha}{N}$. All other transition probabilities are zeros. In each state, the diphone emission probabilities are defined:

$$b_i(k) = \begin{cases} \beta & d_k = s_i \\ \frac{1-\beta}{M} & \text{otherwise} \end{cases}$$

where $b_i(k)$ is a probability of producing diphone d_k at state i , and M is the number of all the diphones in the model. We adjust α and β experimentally, and the flexible adjustment helps to cope with the insertion and deletion errors in the diphone sequences. The Viterbi search with the trie-structured HMM index on the input diphone sequences segments out all the possible morphemes in the given diphone sequence, and enrolls all the segmented morphemes into the triangular table on the proper positions. For example, in figure 6, morphemes such as *ci* (*carry*), *ci-wu* (*delete*),

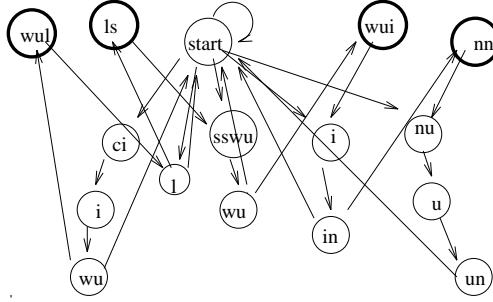


Figure 7: Trie-structured diphone-based HMM index for morphemes *ciwu*, *l*, *swu*, *iss*, *nun*. In each node, if a path from the root (start node) completes a morpheme, a pointer leads to the corresponding morpheme entry in the UPM dictionary. The self-transition for each node is left out except the root for figure simplicity.

l (adnominal verb-ending), *wul* (cry), *swu* (bound-noun) are segmented out and enrolled in the table position (1,2), (1,3), (4,4), (3,4), (5,6). The position (i,j) designates the starting and ending position of each morpheme in the given input *eonjeol*.

The morphotactics modeling is necessary after all the morphemes are enrolled in the table in order to combine only legal morphemes into an *eojeol* (Korean word), and the process is called morpheme-connectivity-checking. Since Korean has well developed postpositions (noun-endings, verb-endings, prefinal verb-endings) which play as grammatical functional morphemes, we must assign each morpheme proper part-of-speech (POS) tags for the efficient connectivity checking. Our more than 400 POS tags which are refined from the 13 major Korean lexical categories are hierarchically organized, and contained in the UPM dictionary (in the name of left and right morphological connectivity, see figure 5). In the case of idiomatic expressions, we place such idioms directly in the dictionary for efficiency, where two different POS tags are necessary for the left and the right morphological connectivity. For single morpheme, the left and the right POS tags are always the same. The separate morpheme-connectivity-matrix (sometimes, it is called morpheme-adjacency-matrix) indicates the legal morpheme combinations using the POS tags defined in the dictionary. So the morphotactics modeling is performed by utilizing two essential components: the POS tags (in the dictionary) and the morpheme-connectivity-matrix. For example, in figure 6, the morpheme *ciwu* (in position (1,3)) can be legally combined with the morpheme *l* (in position (4,4)) to make *ciwu+l* (delete+ADNOMINAL, in position (1,4)) but *ci* cannot be combined with *wul* to make *ci+wul* even if they are in the combinable positions.

The orthographic rule modeling must be integrated with the phonological rule modeling in spoken Korean processing. Since we must deal with the erroneous speech inputs, the conventional rule-based modeling requires so many number of rule applications [13]. So our solution is based on the declarative modeling of both orthographic and phonological rules in a uniform way. That is, in our UPM dictionary, the conjugated verb forms as well as the original verb forms are all enrolled, and the same morphological connectivity information is applied for both original verb forms as well as the conjugated ones. The phonological rule modeling is also accomplished declaratively by having the separate phonemic connectivity information in the dictionary (see figure 5). The phonemic connectivity information for each morpheme declares the possible phonemic changes in the first (left) and the last (right) positioned phonemes in the morpheme, and the phoneme-

connectivity-matrix indicates the legal sound combinations in Korean phonology using the defined phonemic connectivity information. For example, in figure 6, the morpheme *l* can be combined with the morpheme *swu* during the morpheme connectivity checking even if *swu* is actually pronounced as *sswu* (see the input in figure 6). The phoneme-connectivity-matrix supports the legality of the combination of *l* sound with changed *s* to *ss* sound. This legality comes from the Korean phonology rule *glotalization* (one form of consonant dissimilation) stating that *s* sound becomes *ss* sound after *l* sound. In this way, we can declaratively model all the major Korean phonology rules such as (syllable-final consonant) standardization, consonant assimilation, palatalization, glotalization, insertion, deletion, and contraction.

6 Implementation and experiment results

The SKOPE speech and natural language integration architecture was implemented using a standard C and X-window user interface on a UNIX/Sun Sparc platform. The system’s inputs are carefully articulated Korean speeches in a normal laboratory environment, and the outputs are morphologically analyzed *eojeol* sequences which can be directly used by Korean syntactic and semantic analysis modules. We constructed a 1000 morpheme-entry UPM dictionary in a UNIX operating system domain [14], and built morpheme connectivity and phoneme connectivity matrices for the phonological/morphological co-analysis. The UPM dictionary is indexed using the diphone transcribed HMM headers for each morpheme, which are organized into a trie. Since we don’t have any standard segmented Korean speech database yet, we constructed our own by recording and manually segmenting 73 most frequent Korean diphones. The 73 diphones are acquired from the 300 Korean *eojeols* (each *eojeol* is pronounced 15 times by a female speaker) in 50 Korean sentences which can appear in natural language commanding to the UNIX operating system [14].

Several experiments were performed to verify the system’s performance of time-shift invariance, diphone recognition, and final *eojeol* recognition including the spoken language morphological analysis. In each experiment, the input speech patterns were prepared as follows: *eojeols* were recorded in a normal laboratory environment with an average S/N ratio of 12 dB. Speech data were sampled at 16kHz-16bit, and hamming-windowed. From this windowed data, 512-point DTFTs were computed at 5 msec intervals. The DTFTs were used to generate 16 Mel-scale filter-bank coefficients at 10 msec frame size [8]. These spectra were normalized to produce suitable input levels for the four-layer TDNNs. We used hyperbolic arc tangent error function for the weight updating [15] in the back propagation training, and updated the weights after a small number of iterations [16].

6.1 Time-shift invariance of Korean diphones

We generated 2400 diphone samples for typical 12 Korean diphones. The input patterns for two test cases are set the same in order to compare the *no-time-shift* and *time-shift* cases. Figure 8 shows that the Korean diphone recognition maintains the time shift invariance property of TDNN and suggests the optimal time interval near 200 - 250 msec.

6.2 Comparison of diphone recognition vs. phoneme recognition

This experiment is to show that diphones can improve the recognition rate of Korean vowels regardless of many rising diphthongs compared with the phoneme recognition. In the test, we set

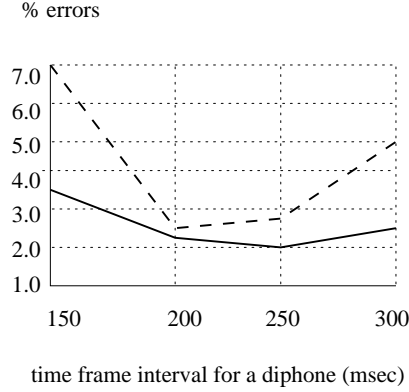


Figure 8: Average error rate of the segmented time frame (solid lines) versus the same time frame with maximum 40 msec left or right temporal shift (dotted lines)

unit of recognition	number of targets	number of samples	recognition rate
phoneme	9	1080	94.06%
	17	2040	89.80%
diphone	9	1080	95.42%
	17	2040	95.27%

Figure 9: Diphone recognition versus phoneme recognition test

150 msec time range for the phoneme and 200 msec for the diphone segmentation. Compared with the phoneme recognition, figure 9 shows that diphone recognition performance doesn't drop much when the number of targets with similar features doubly increases.

6.3 Performance of continuous diphone recognition

In this experiment, we pronounced carefully chosen 66 *eojeols* 15 times to generate about 5500 diphone patterns for training. The 5500 training samples are used to train the vowel group TDNN and 18 different sub-TDNNs for each diphone group. During the recognition, the new 262 *eojeols* are selected to generate the test patterns of 2432 *eojeols*, and these test patterns are shifted 30 msec during the recognition to obtain the TDNN diphone spotting performance in a continuous speech. Figure 10-a shows the continuous diphone spotting performance. We have total 7772 target diphones from the 2432 test *eojeol* patterns. The *correct* designates that the correct target diphones were spotted in the testing position, and the *delete* designates the other case (including the substitution errors). The *insert* designates that the non-target diphones were spotted in the testing position. To compare the ability of handling the continuous speech, we also tested the diphone spotting using the hand segmented test patterns with the same 7772 target diphones. Figure 10-b shows the segmented diphone recognition performance. Since the test data are already hand-segmented before input, there are no insertion and deletion errors in this case. The fact that

a. continuous diphones				
	total	correct	delete	insert
pattern size (rec. rate)	7772	7259 (93.3%)	513 (6.4%)	3000

b. segmented diphones			
	vowel group	sub-TDNNs average	total average
rec. rate	94.9%	98.8%	93.8%

Figure 10: Continuous diphone spotting versus segmented diphone recognition

the segmented speech performance is not much better than the continuous one (93.8% vs. 93.3%) demonstrates the diphone’s suitability to handling the Korean continuous speech.

6.4 Performance of continuous speech morphological analysis

In order to test the ability of full *eojeol* recognition including the Viterbi-based lexical decoding and phonological/morphological co-analysis, a middle-vocabulary experiment was carried out. The task is a speaker-dependent and continuous *eojeol* recognition which produces the morphologically analyzed *eojeol* sequences directly from the speech inputs. In the process, the speech recognizer produces the erroneous diphone sequences in input *eonjeols*, and then the Viterbi morphological analyzer segments them with the error correction and produces the final analyzed *eojeols*. So, in this task, all the intermediate steps, that is, diphone spotting, lexical decoding and morphological/phonological analysis, are combined to produce the final recognition performance. The same 328 *eojeols* in section 6.3 were fed to the SKOPE integration architecture that has the pre-trained TDNNs (with 66 *eojeols*). Figure 11-a shows the performance with the trained 66 *eojeols* and figure 11-b shows the final performance of the total 328 *eojeols*. We have total 4266 target morphemes from the same 328 *eojeols* used in section 6.3. In the figure, the *correct* designates that the correct morpheme sequences can be analyzed from the speech input, and the *delete* means that the correct morpheme sequences cannot be generated (including the substitution errors). The *insert* designates the percentage of the spurious morphemes that are generated from the insertion errors. The performance is about 80.6% correctness in the final morphological analysis with the mostly untrained new data, which is quite promising considering the complexity of the task.

7 Comparison with related researches

Recently, the idea of sending only n best speech recognition results to a natural language module has been implemented using the time-synchronous Viterbi-style search algorithm [1]. The algorithm was also improved by the word-dependent search [2] and by adding the A* backward tree search [17]. The n-best integration scheme has been mostly utilized for HMM-based continuous speech recognition systems, and many existing speech systems and natural language systems were successfully integrated using the n-best word search techniques [3, 4]. However, until now, the n-best search techniques are only implemented to directly produce the n-best sentences using the word sequences, and this word-level integration is inefficient for the morphologically complex languages

a. for trained 66 eojeols

	total	correct	delete	insert
pattern size (rec. rate)	2083	1847 (88.67%)	236	311

b. for total 328 eojeols

	total	correct	delete	insert
pattern size (rec. rate)	4266	3440 (80.64%)	826	902

Figure 11: Continuous speech input morphological analysis performance

such as Korean. On the contrary, our integration is at the morpheme-level directly decoding the PLU sequences with the morphological processing because we need more sophisticated phonological/morphological handling in the early stage of the integration process. The word-level n-best integration also assumes the word-level dictionary which is an unreasonable assumption for morphologically complex languages. According to the Harper and others’ recent classification [18], n-best integration is a typical loosely-coupled example.

The HMM-LR integration [19, 20] was implemented using the HMM’s phoneme spotting ability integrated with the generalized LR parsing techniques [21]. Unlike the n-best integration, the HMM-LR integration was more tight and implemented at the phoneme-level by extending the LR parser’s terminal symbols to cover the phonetic transcriptions. In this scheme, the LR parsing selects the most probable parsing results by obtaining the probability of the end-point candidate phonemes from the HMM’s forward probability calculation. So the total integrated system is working by the LR parser’s prediction of the next phoneme candidates which are then verified by the HMM’s phoneme spotting abilities. The idea of extending the LR grammar to the phonetic transcriptions seems to be working for the phoneme-level integration. However, the scheme doesn’t have any separate language-level dictionary, which results in the degenerated phonological/morphological processing, and also suffers from difficulty in the necessary scale-ups. On the contrary, our SKOPE integration architecture focuses on the general phonological/morphological handling during the integration which is essential for the agglutinative languages. The idea of extending LR grammar to the phonetic transcriptions was also applied to the TDNN-LR integration method [22, 23] which was similarly implemented by replacing HMM’s phoneme spotting by the TDNN’s phoneme spotting. The integration was implemented by dynamic time warping (DTW) level-building search [24] between TDNN’s phoneme sequences and LR grammar’s phoneme sequences. However, the performance was relatively poor compared with the HMM-LR integration method [22]. There are basically two reasons for the poor TDNN-LR performance compared with the HMM-LR integration: 1) the TDNN model has rarely been applied to the practical large vocabulary systems yet, therefore it lacks the fine tuning compared with the popular HMM models, and 2) the TDNN model has yet to find a right way to be effectively integrated into the natural language processing model. The HMM model supports a natural integration into the general chart-based parsing models such as generalized LR parsing because there are well-defined probabilistic

search techniques in the language as well as in the speech levels. However, output activations of the multiple TDNNs are difficult to normalize and therefore difficult to be naturally integrated into the popular probabilistic search schemes such as Viterbi search. Our SKOPE architecture adopts Viterbi search with pre-defined transition and emission probabilities, and use the Viterbi search for only segmenting erroneous dipphone strings. All the other morphological processing steps are generally performed according to the symbolic natural language processing model.

The more tightly-coupled systems have also been researched to integrate all the knowledge sources of spoken language processing from acoustic to semantic into a single interdependent model that cannot easily be separated. In these systems, syntactic parser directly deals with acoustic-level inputs. For example, Ney [25] extended CYK parsing algorithm to cover acoustic inputs by exhaustively finding all possible endpoints for every terminal symbol. In the similar vein, the HMM can be extended to handle recursive embedding for context-free grammar processing [26]. However, these acoustic-level syntactic parsers are computationally expensive since the parsing complexity is at best $O(n^3)$ where n could be in several hundreds when the parsers directly deal with the speech frames. The SKOPE integration is tighter than loosely-coupled n -best techniques, but less tight compared with these tightly-coupled systems. We agree that the high-level linguistic constraints should restrict the underlying speech recognition in some ways as in the tightly-coupled systems, but disagree that the constraints should be in a syntax level. The more tightly-coupled systems are often impractical for large-scale spoken language processing because of the time complexity. Moreover, we still don't have much knowledge to tell how much top-down feedback is actually helpful to improve the speech recognition process. As an engineering point of view, semi-tightly-coupled systems are quite feasible for large complex systems under the current technology. In this regard, SKOPE project adopts a semi-tightly-coupled integration technique between speech and language processing, especially morphological processing.

8 Conclusions

This paper presents a morpheme-level integration architecture of speech and natural language in a connectionist continuous speech recognition model for agglutinative languages such as Korean. Our main contributions are to present the morphologically conditioned semi-tight integration model that can support sophisticated phonological/morphological processing in the integration of speech and language, which is essential for morphologically complex agglutinative languages. Also, the SKOPE integration architecture is a first attempt to develop a morphologically general integration model using the connectionist speech recognition paradigm.

The SKOPE speech and language integration architecture has many novel features for speech and natural language processing. First, the dipphone-based TDNN proposes a nice sub-word unit of recognition, well reflecting the Korean phonetic characteristics. Secondly, the morphological analysis combined with the declarative phonological rule modeling is well suited to the phonetic spelling into the orthographic morpheme mapping, which is an essential task for every spoken language processing model. Finally, the trie-structured HMM indexing for UPM dictionary enables the Viterbi style search to be applied to the thorny morpheme segmentation and lexical decoding problem, and also provides natural integration of symbolic natural language processing techniques with probabilistic decoding schemes. The experiments show that the final morphological analysis performance from continuous speech is over 80.6% in a middle-vocabulary speaker-dependent recognition task, which is very promising in considering the continuous speech and the combination of several steps of

performances such as diphone spotting, lexical decoding and morphological/phonological analysis. Since the integration architecture is based on general linguistic notion of phoneme and morpheme, the architecture is not restricted to Korean. The SKOPE architecture can be extended to any agglutinative language which has clear-cut morphological boundaries such as Japanese, and possibly to other Indo-European languages which exhibit well-developed morphological phenomena such as German. We are now extending the integration technique to Japanese.

Acknowledgements

This research was partly supported by grants from KOSEF and ETRI in Korea. Many of our former or current students implemented the idea and we thank all of them. Especially, we thank Kyunghee Kim for implementing diphone-based Korean speech recognizer, WonIl Lee for coding the lexicon and the morphological parser, and finally ByungChang Kim for implementing Viterbi search for the diphones. Professor Hong Jeong directed us to the deep world of signal processing for the earlier part of the project.

References

- [1] Y. Chow and R. Schwartz, "The n-best algorithm: An efficient procedures for finding top N sentence hypothesis," in *Proceedings of the second DARPA workshop on speech and natural language*, Los Altos, CA, 1989, Morgan Kaufmann Publishers, Inc.
- [2] R. Schwartz and S. Austin, "Efficient, high-performance algorithms for n-best search," in *Proceedings of the third DARPA workshop on speech and natural language*, Los Altos, CA, 1990, Morgan Kaufmann Publishers, Inc.
- [3] M. Agnas, H. Alshaw, I. Bretan, D. Carter, K. Ceder, M. Collins, R. Crouch, V. Digalakis, B. Ekholm, B. Bamback, J. Kaja, J. Karlgren, B. Lyberg, P. Price, S. Pulman, M. Rayner, C. Samuelsson, and T. Svensson, "Spoken language trnaslator: first year report," Technical Report ISRN SICS-R-94/03-SE, Swedish Institute of Computer Science and SRI International, 1994.
- [4] M. Bates, R. Bobrow, P. Fung, R. Ingria, F. Kubala, J. Makhoul, L. Nguyen, R. Schwartz, and D. Stallard, "The BBN/HARC spoken language understanding system," in *Proceedings of the ICASSP-93*, 1993.
- [5] G. Forney, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, pp. 268–278, 1973.
- [6] K. Kim, G. Lee, and J. Lee, "Integrating TDNN-based diphone recognition with table-driven morphology parsing for understanding of spoken Korean," in *Proceedings of the international conference on spoken language processing (ICSLP)*, 1994.
- [7] D. Morgan and C. L. Scofield, *Neural networks and speech processing*, Kluwer Academic Publishers, Inc., 1991.
- [8] A. Waibel, T. Hanaazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 3, pp. 328 – 339, 1989.

- [9] W. Lee, G. Lee, and J. Lee, "Table-driven neural syntactic analysis of spoken Korean," in *Proceedings of COLING-94*, 1994.
- [10] K. F. Lee, *Automatic speech recognition*, Kluwer Academic Publishers, Inc., 1989.
- [11] R. Sproat, *Morphology and computation*, The MIT Press, 1992.
- [12] A. Aho and J. D. Ullman, *The theory of parsing, translation, and compiling, Vol 1: parsing*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [13] K. Koskenniemi, "Two-level model for morphological analysis," in *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, 1983.
- [14] W. Lee and G. Lee, "From natural language to shell-script: A case-based reasoning system for automatic unix programming," *Expert systems with applications: An international journal*, vol. 9, no. 1, pp. 71–79, 1995.
- [15] S. E. Fahlman, "Faster-learning variations on back-propagation: An empirical study," in *Proceedings of the 1988 connectionist models summer school*, 1988.
- [16] P. Haffner, A. Waibel, H. Sawai, and K. Shikano, "Fast back-propagation learning methods for large phonemic neural networks," in *Proc. of the Eurospeech-89*, 1989.
- [17] F. K. Soong and E. Huang, "A tree-trellis based fast search for finding the n-best sentence hypotheses in continuous speech recognition," in *Proceedings of the third DARPA workshop on speech and natural language*, Los Altos, CA, 1990, Morgan Kaufmann Publishers, Inc.
- [18] M. Harper, L. Jamieson, C. Mitchell, G. Ying, S. Potisuk, P. Srinivasan, R. Chen, C. Zoltowski, L. McPheters, B. Pellom, and R. Helzerman, "Integrating language models with speech recognition," in *Proceedings of the AAAI'94 workshop on integration of natural language and speech processing*, 1994.
- [19] K. Kita, T. Kawabata, and H. Saito, "HMM continuous speech recognition using predictive LR parsing," in *Proceedings of the ICASSP-89*, 1989.
- [20] T. Hanazawa, K. Kita, S. Nakamura, T. Kawabata, and K. Shikano, "ATR HMM-LR continuous speech recognition system," in *Proceedings of the ICASSP-90*, 1990.
- [21] M. Tomita, *Efficient parsing for natural language - A fast algorithm for practical systems*, Kluwer Academic Publishers, 1986.
- [22] H. Sawai, "TDNN-LR continuous speech recognition system using adaptive incremental TDNN training," in *Proceedings of the ICASSP-91*, 1991.
- [23] H. Sawai, "The TDNN-LR large-vocabulary and continuous speech recognition system," in *Proceedings of the international conference on spoken language processing (ICSLP)*, 1990.
- [24] C. Myers and L. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. on ASSP*, vol. 29, no. 2, pp. 284–279, 1981.

- [25] H. Ney, “Dynamic programming parsing for context-free grammars in continuous speech recognition,” *IEEE Trans. on Signal Processing*, vol. 39, no. 2, , 1991.
- [26] K. Lari and S. Young, “Applications of stochastic context-free grammars using the inside-outside algorithm,” *Computer speech and language*, vol. 5, no. 3, , 1991.