Table-driven Neural Syntactic Analysis of Spoken Korean

WonIl Lee, Geunbae Lee, Jong-Hyeok Lee

Computer Science Deptartment of POSTECH. KOREA
Tel: +82-562-279-2254, Fax: +82-562-279-2299

Abstract

A CYK-table-driven interactive relaxation parsing method of spoken Korean, integrated with the CYK-based morphological analysis is introduced. An extension of the Categorial Grammar is introduced to treat the free word-order in Korean. The table-driven control of interactive relaxation gives efficiency in constituent searching and expectation generation. The lexical nature of the Categorial Grammar and the distributed nature of the interactive relaxation parsing together show a smooth integration of both bottom-up and top-down effects during the spoken language analysis.

1. Introduction

Spoken language input may be one of the most preferred interface with computer systems if speech can be well integrated with the natural language processing.

There have been a few attempts to integrate speech recognition with the natural language understanding. (Hayes 1986) adopted the case frame instantiation technique to parse continuously spoken English sentences given in the form of a word lattice (a set of word candidates hypothesized by a speech recognition module) and produced a frame representation of the utterence.

(Poesio 1987) suggested a modified caseframe parsing to parse a word lattice in Italian. (Lee,, L. S. 1987) developed a prototype Chinese (Mandarin) dictation machine which takes a syllable lattice and produces a Chinese character sequence. (Saito 1991) used GLR technique to parse a sequence of Japanese phonemes and, integrated with the HMM technique, developed a HMM-LR parser (Kita 1991).

However, spoken Korean has many distinguished characteristics compared with spoken English and even with spoken Japanese, for example, comlpex word formation, phonological changes and irregular conjugations. In this paper, we present a CYK-table driven interactive relaxation parsing method of spoken Korean, integrated with the CYK-based morphological analysis. A Korean phoneme lattice is assumed as input and the CYK-based morphological analysis copes with the difficulties of complex word formation, phonological changes and irregular conjugations. Categorial Grammar is adapted and extended to treat the free word-order in Korean and the CYK-table driven interactive relaxation parsing gives efficiency beyond the general interactive relaxation parsing methods such as (Howells 1988; Nijholt 1990).

2. Features of spoken Korean

Korean, a SOV and an agglunating language, has the following characteristics:

1) A Korean word, Eojeol, consists of more than one morphemes with clear-cut boundaries in between. For example, an Eojeol "*pha-il-dul-ul*(files[obj])" consists of 3 morphemes:

pha-il	=	*il*	+	*dul*	+	*ul*
files[obj]		file		[pl. suffix]		[obj marker]

.

2) Korean is a postpositional language with noun-endings, verb-endings and prefinal verb-endings. These functional morphemes determine the noun's case role, verb's tense, modality, and modification relations between phrases. For example, in "*e-cey* ***swu-ceng-ha-yess-ten*** *pha-il* (the file **that was edited** yesterday)" the verb "*swu-ceng* (edit)" is of past tense and modifies "*pha-il* (file)":

su-ceng-ha	*yess*	*ten*	*pha-il*
edit	[past]	[adnominal]	file

3) Korean has relatively free word order compared to SVO languages, such as English. For example, the sentence "*nay-ka e-cey swu-ceyng-ha-yess-ten pha-il-tul-ul /tmp lo pok-sa-ha-ye-la* (Copy the files that was edited by me yesterday to /tmp.)" may be written as
"*e-cey nay-ka swu-ceng-ha-yess-ten ...* " or
"*/tmp lo e-cey nay-ka swu-ceng-ha-yess-ten ...* ."

Besides these characteristics of written Korean, spoken Korean has the following more characteristics:

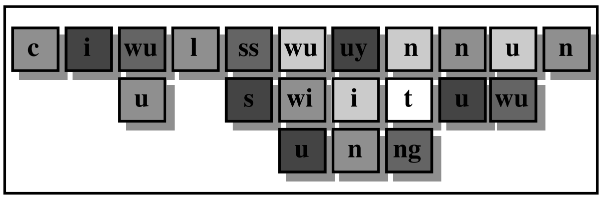
4) The unit of pause in a speech (Eonjeol) may be different from that of a written text (Eojeol). For example, in speaking "*nay-ka e-cey swu-ceng-ha-yess-ten pha-il-tul-ul /tmp lo pok-sa-ha-ye-la* (spaces delimit Eojeols), a person may pause after "*nay-ka*" and after "*e-cey swu-ceng-ha-yess-ten pha-il-tul-ul*" and after "*/tmp lo pok-sa-ha-ye-la.*"

5) Phonological changes occur in a morpheme, between morphemes in an Eojeol, and between Eojeols in an Eonjeol. These changes include assimilation, dissimilation, and contraction. For example, a morpheme "*pok-sa*" is pronounced as /*pok-ssa*/ and "*yess*" is pronounced as /yet/. An Eojeol "*su-ceng-ha-yess-ten*" is pronounced as /su-ceng-ha-yet-tten/.

3. Table driven neural syntactic analysis

This section explains interactive relaxation parsing of spoken Korean using neural network, its underlying grammar, and control mechanism.

A sequence of candidate phonemes in phoneme lattice [figure 1] is assumed to be the output of the speech recognizer.



[Figure 1. Phoneme lattice]

A CYK-based morphological analyzer is used to extract a morpheme lattice from the phoneme lattice. In the morphological analysis, special procedural attachments resolve the phonological changes. The use of phoneme lattice gives the problem of exponential number of Eonjeol candidates. For this problem, trie data structure is used for the phonetic transcription-to-orthographic morpheme dictionary (morpheme-level phonetic dictionary).

3.1. Extending the Categorial Grammar

To model the syntax of Korean, we extended the Categorial Grammar in two ways (Zeevat 1988; Uszkoreit 1986).

A (directional) Categorial Grammar is an ordered quintuple G = <V, C, S, R, f>, where

1) V: the vocabulary set,

2) C: a finite set of basic categories which generates a full set C' of categories via recursive application of the following category formation rules:
if $a \in C$, then $a \in C'$ and
if $a \in C'$ and $b \in C'$, then $a/b \in C'$ and $a\backslash b \in C'$,

3) S: the category for sentences,

4) R: a set of functional application rules such as
left cancellation : $A \quad B\backslash A \dashrightarrow B$
right cancellation : $B/A \quad A \dashrightarrow B$

5) f: an assignment function from elements of V into the subsets of C'.

To treat the free word-order in Korean, we extended the category formation rules and the application rules:

2') Extended category formation rules:
if $a \in C$, then $a \in C'$ and
if $a \in C'$ and $S \subset C'$, then $a/S \in C'$ and $a\backslash S \in C'$ and

4') Extended functional application rules :
left cancellation :
$Ai \quad B\backslash\{A1,...,An\} \dashrightarrow B\backslash\{A1,...,Ai\text{-}1,Ai+1,...,An\}$
right cancellation :
$B/\{A1,...,An\} \quad Ai \dashrightarrow B/\{A1,...,Ai\text{-}1,Ai+1,...,An\}$

3.2. Interactive relaxation parsing

(Howells 1988) developed an interactive relaxation parsing method which used a dynamic network building scheme, and decay over time with competition instead of explicit inhibitory links, which is similar to the (Reggia 1987)'s approach.

The interactive relaxation algorithm consists of the following steps (Howells 1988): 1) add nodes, 2) spread activation and 3) decay. Bottom-up information gathering and top-down expectations occur during the parsing.

1) to add a node:

A grammar node is added for each sense of morphemes when the parsing begins. Statistical information on the senses of a morpheme determines the initial activation value of the senses.

A grammar node which has more activation than the predefined threshold Θ makes new nodes (expectations). The newly generated nodes represent candidate parse trees containing the generator node.

2) to spread activation:

A predefined portion, P, of a node's total activation, A, is passed upward, to bigger parse trees. When more than one destination nodes exist, they compete to get more activations. A higher node with activation a_i gets the following amount of activations:

$$A \times P \times \frac{ai^2}{\sum_{j=1}^{n} aj^2}$$

A higher level node (with total activation A) spreads a predefined portion (Q) of its activation value equally to the constituents. When there are n constituents, a constituent gets the following amount of activations:

$$A \times Q \times \frac{1}{n}$$

3) to decay:

A node's activation value (A) after decay is A times (1 - D), where D is the decay ratio. Moreover, a node with less constituents than needed is penalized by the number of actual constituents (C_a) divided by the number of required constituents (C_r). After all, a node's activation value changed to

$$A \times (1 - D) \times \frac{Ca}{Cr}$$

And finally a node whose activation value is less than the predefined threshold Φ is removed.

3.3. CYK-table-driven control

The interactive relaxation parsing scheme (Howells 1988) lacks efficient control structures for constituent searching and expectation generation. We provided the positional information through the CYK-table and the structuring information through the Categorial Grammar formalism. Using Categorial Grammar makes the parse tree be a binary tree, not a general n-ary tree.

All the grammar nodes reside in a CYK-table. The position (i,j) in the table explicitly says where to find the constituents, where to add new expectations and what the node there stands for.

A node in CYK(i,j) represents a parse tree for the input segment from i to j. A node in CYK(i,j) with category P, called $P_{(i,j)}$, can be used in 3 ways to construct larger parse trees:

1) $Q/P_{(k,i-1)}$ $P_{(i,j)}$ --> $Q_{(k,j)}$ or
$P_{(i,j)}$ $Q\backslash P_{(j+1,k)}$ --> $Q_{(i,k)}$.

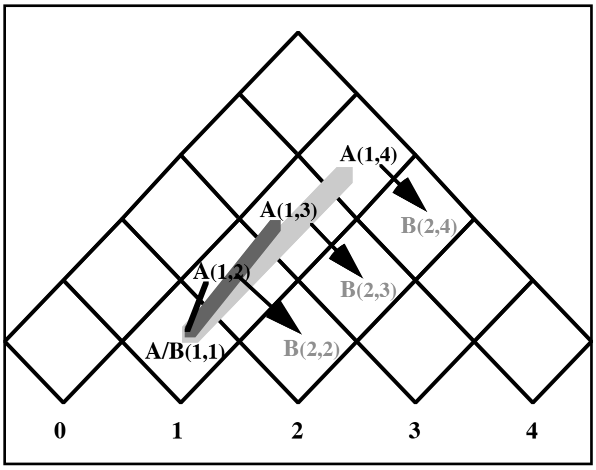
$P_{(i,j)}$ combins with $Q/P_{(k,i-1)}$ for some k < i and becomes a constituent of $Q_{(k,j)}$. Or it combins with $Q\backslash P_{(j+1,k)}$ for some k > j and becomes a constituent of $Q_{(i,k)}$. In these two cases $P_{(i,j)}$ is used as an argument of the functional categories.

2) $A/B_{(i,j)}$ $B_{(j+1,k)}$ --> $A_{(i,k)}$.

$P_{(i,j)}$ combines $B_{(j+1,k)}$ for some k > j and creates a larger parse tree $A_{(i,k)}$, when $P = A/B$. In this case, $P_{(i,j)}$ is a functional category A/B and searches for an argument B on the right side.

3) $B_{(k,i-1)}$ $A\backslash B_{(i,j)}$ --> $A_{(k,j)}$.

$P_{(i,j)}$ combines $B_{(k,i-1)}$ for some k < i and creates $A_{(k,j)}$, when $P=A\backslash B$. In this case, $P_{(i,j)}$ is a functional category $A\backslash B$ and searches for an argument B on the left side.



[Figure 2. $A/B_{(1,1)}$'s expectation generation]

The following scenarios can explain the CYK-table driven interactive relaxation parsing of Korean.

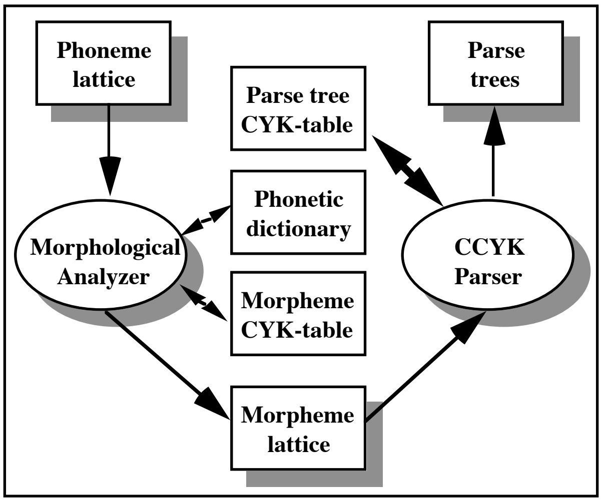
A node $A/B_{(i,j)}$ whose activation value is greater than the threshold Θ makes new nodes ($A_{(i,k)}$'s for all j < k < input-length), and each $A_{(i,k)}$ looks for a constituent $B_{(j+1,k)}$. [Figure 2] shows $A/B_{(1,1)}$'s expectation generations. There are 3 possible A's which can have $A/B_{(1,1)}$ as their constituents: $A_{(1,2)}$, $A_{(1,3)}$ and $A_{(1,4)}$. $A_{(1,2)}$ looks for $B_{(2,2)}$, $A_{(1,3)}$ for $B_{(2,3)}$, and $A_{(1,4)}$ for $B_{(2,4)}$.

Symmetrically, a node $A\backslash B_{(i,j)}$ whose activation value is greater than the threshold Θ makes new nodes ($A_{(k,j)}$'s for all 0 < k < i) and each $A_{(k,j)}$ looks for a constituent $B_{(k,i-1)}$.

Only the nodes with enough activations generate hypotheses, and the hypotheses which lack of constituents disappear rapidly by the decay with penalty mechanism. Each node (hypothesis) which looks for a constituent drives the parsing, and the efficiency of the constituent searching is guaranteed by the CYK-table. We call the parsing technique as Connectionist-CYK parsing.

4. System architecture

The Connectionist-CYK parser is incorporated into the DINX (Dialog Interface to UNIX) system which is under development in POSTECH (Lee, W. I. and Lee, G.B. 1993). [Figure 3] shows the part of DINX system architecture. There are two modules in the system (for language analysis): morphological analyzer and the CCYK parser.



[Figure 3. Speech parser architecture]

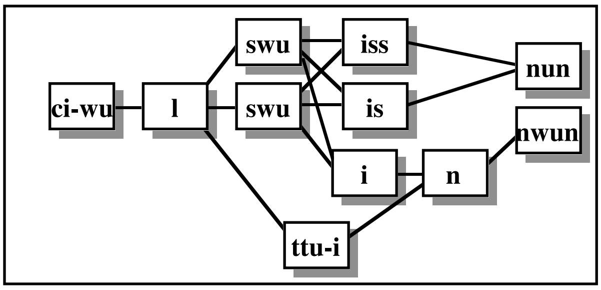
A phoneme lattice for each Eonjeol is analyzed by an extension of the CYK-based morphological analyzer (Lee, E. C. 1992). The result of the analysis is a morpheme lattice (see [Figure 4]) for each Eonjeol. These morpheme lattices are linked, and the morpheme lattice for entire sentence is constructed. Grammar nodes for the senses of each morpheme are created in the CYK-table and the following four steps are repeated for a fixed number of iterations: 1) making hypotheses (new nodes), 2) constituent searching, 3) computing out-going activations and 4) updating activation values. Functional category nodes $^{A}/_{B}(i,j)$ and $^{A}\backslash_{B}(i,j)$ whose activation values are greater than the threshold generate the expectations $^{A}(i,k)$'s and $^{A}(k,j)$'s respectively. All the hypotheses (nodes which lack of a constituent) search for their constituents. Outgoing bottom-up and top-down activations are computed for each node. Each node gathers incoming activations and decays.

5. Sample run

In this section, a detailed example of the Connectionist-CYK parsing is given. The system parameters are as follows: bottom-up activation ratio P = 0.87, top-down activation ratio Q = 0.39, decay ratio D = 0.495, expectation threshold Θ = 6.66 and remove threshold Φ = 0.66.

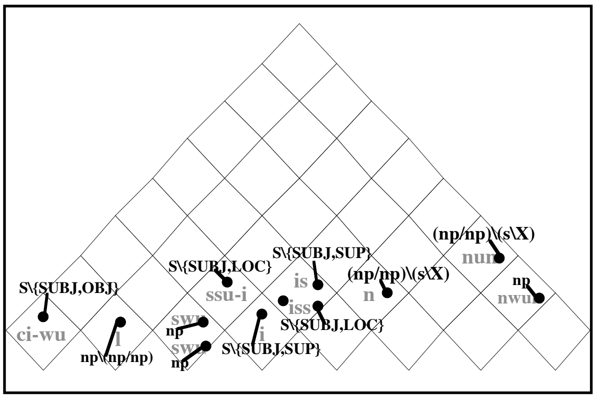
The sentence "*ci-wul su iss-nun pha-il-dul-ul po-ye-la* (List the files which can be removed.)" is assumed to be spoken with two pauses, "*ci-wul su iss-nun / pha-il-dul-ul / po-ye-la.*" [Figure 1] showed the first phoneme lattice of the three.

The phoneme lattices are analyzed one by one, and a morpheme lattice for the sentence is created by merging the morpheme lattice for each Eonjeol [Figure 4].

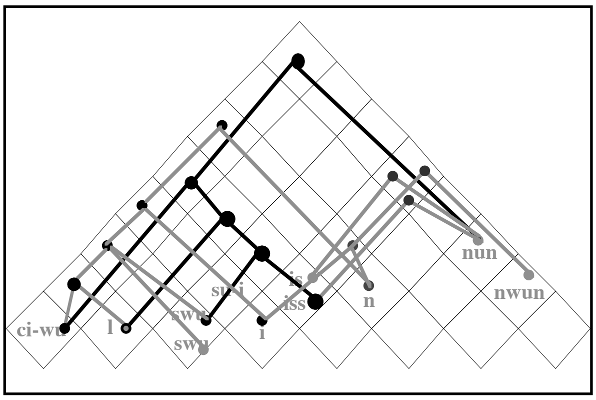


[Figure 4. A morpheme lattice]

Grammar nodes for the senses of each morpheme are created [Figure 5 (a)].



(a) Initialization of the CYK-table



(b) After 6-th iteration

[Figure 5. Sample run]

Each dot represents a node and the darkness of a dot denotes the degree of activation. The morpheme lattice is embedded in the CYK-table, and the senses of each morpheme are created in the corresponding position in the table. Each node with functional category generates hypotheses, and each hypothesis searches for their constituents. After the 1-st iteration, 209 nodes are in the table. The number of nodes change to 282, 302, 289, and 265 along the iterations. After 6-th iteration, the number of nodes decreases to 253, and the correct parse tree for the whole morpheme lattice is created ([Figure 5 (b)] shows a part of the parse tree). After the 7-th iteration, the number of nodes decreases 191, 180, 163, 103, 98, ..., 78. After 30-th, the correct parse tree which covers the entire sentence is extracted.

6. Conclusions and future works

In this paper we have developed a CYK-table-driven interactive relaxation parsing method (CCYK parsing) of spoken Korean, integrated with the CYK-based morphological analysis.

We have shown that the partially free word order and the postpositions are easily and naturally encoded by the Extended Categorial Grammar. The use of Categorial Grammar with the interactive relaxation parsing results in a lexical-based, locally distributed and island-driven parsing. A node generates hypotheses if it gathers enough bottom-up activation (evidence) and each hypothesis locally searches for constituents. Decay with penalty removes failed and incorrect hypotheses.

The use of the phoneme lattice and the morpheme lattice removes the redundancy in postprocessing and parsing of spoken Korean, and makes the whole interactions among the phonemes and morphemes possible. The activation value of parses gives a clue to the selection of the best parse. The statistical information between senses of morphemes can make the selection more reliable. As noted in (Howells 1988), changing the system parameters gives the flexibility of retaining the alternatives. We can have all the parses in one setting and can have only the best parse on the another setting depending on the application. We plan to add the unification ability for semantic analysis to our work, and finally develop a connectionist parsing method for the full Categorial Unification Grammar.

References

Hayes, P. J. , Hauptmann, A. G. et al. (1986). Parsing spoken language: A semantic caseframe approach. In *Proc. of COLING'86*, Bonn, UK.

Howells, T. (1988). A connectionist parser. In *Proceedings of the tenth Annual Conference of the Cognitive Science Society*, pp18-25.

Kita, K., Kawabata, T. and Saito, H. (1991). GLR Parsing in Hidden Markov Model. In M. Tomita (eds.) *Generalized LR Parsing* , Kluwer.

Lee, E. C. and Lee, J. H. (1992). The implementation of Korean Morphological Analyzer Using Hierarchical Symbolic Connectivity Information. In *Proceedings of the 4th Conference of Korean and Korean Information Processing* , pp95-104, (in Korean).

Lee, L. S., Tseng, C. Y. et al. (1987). The preminary results of a Mandarine dictation machine based upon Chinese natural language analysis. In *Proc. of IJCAI'87*, Milan.

Lee, W. I. and Lee, G. B. (1993). From Natural Language to Shell-script: A Case-based Reasoning System for Automatic UNIX Programming. In Proceedings of the First Korea-Japan Joint Conference on Expert System.

Nijholt, A. (1990). Meta-parsing in neural networks. In *Proceedings of the Tenth European Meeting on Cybernetics and Systems Research*, Austin Society for Cybernetics Studies, pp969-976, Vienna.

Poesio, M. and Rullent, C. (1987). Modified caseframe parsing for speech understanding systems, In *Proc. of IJCAI'87*, Milan.

Reggia, J. A. (1987). Properties of a competition-based activation mechanism in neuromimetic network models. In *Proceedings of the IEEE First International Conference on Neural Networks*, II, San Diego, CA.

Saito, H. and Tomita, M. (1991). GLR Parsing for Noisy Input. In M. Tomita (eds.) *Generalizde LR Parsing* , 1991, Kluwer.

Uszkoreit, H. (1986). Categorial Unification Grammars. In *Proceedings of the COLING'86*, pp187-194.

Zeevat. H. (1988). Combining Categorial Grammar and Unification. In U. Reyle and C Rohrer (eds.) *Natural Language Parsing and Linguistic Theories*, pp202-209.