Predictive Morphological Analysis of Korean with Dynamic Programming

Deok-Bong Kim, Key-Sun Choi and Kang-Hyuk Lee
Center for Artificial Intelligence Research
Department of Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
E-mail: {dlokim, kschlie, kkhlee} @cs.kaist.ac.kr

Abstract

In this paper, we present an efficient morphological analysis model for Korean which produces from an input word all the feasible sequences of morphemes in the word. This model is deterministic in applying spelling rules, and has few redundant computations in processing complex and ambiguous words. This is the effect of three types of new techniques: first, a new method for interpreting spelling rules; second, predictive rule applications which restrict to the spelling rules suitable for the input word; third, the use of dynamic programming which enables the analyzer to avoid recomputing analyzed substrings in case the input word is morphologically ambiguous. Our model has been experimented with 413,975 words randomly selected from the corpus of Korean elementary textbooks. Experimental results show that our model guarantees fast and reliable processing.

※ 본 논문은 과학제단 목적기조연구 “Robust 자식기반시스템 구축기술에 관한 연구”의 세부과제 “자연언어 인터페이스를 위한 도구 환경의 연구개발”로서 과학제단의 부분적인 지원을 받은 결과이다.
1. Introduction

Morphological analysis has been recognized to be very crucial in processing languages like Finnish, Japanese, and Korean that are highly agglutinative in nature (Koskenniemi (1983), Sproat (1992), Barton et al. (1987), among others). Being so important, much of NLP research for Korean has focused on efficient and reliable morphological processing. For example, Kang and Kim (1991) and Kwon and Choe (1991) proposed a dictionary-based approach which mainly depends on a large ending-dictionary containing all the possible allomorphs and compound endings. With these approaches, however, it is extremely difficult to develop sharable morphological rules (i.e., spelling rules and morphotactic rules) because they are entirely hard-coded in the system. Lee (1992) is an effort to develop a Korean version of two-level morpholoty. The two-level approach to Korean morphological processing is vulnerable to the problems of massive backtracking and recurrent computations (Barton et al. (1987), Sproat (1992)). Kim (1987) presented a morphological chart parsing method based on the CYK algorithm (Aho(1972)) to avoid the work of recomputing analyzed substrings. The CYK parsing approach to morphology has a potentially serious problem in analyzing highly inflected words, because it is difficult to compute triangular parse tables for such words which are the essence of the analyzer (cf. Aho (1972, 314-320)).

In this paper, we propose an efficient morphological analysis model for Korean which has few redundant computations. Our model also provides
the rule formalism in which morphological rules can be easily described. Though our model has been initially developed for Korean morphological analysis, it is also applicable and extendible to other languages, especially agglutinative languages such as Japanese and Finnish. In section 2, it is presented how morphological knowledge is described in our model. Section 2.1 presents the Korean alphabet and syllable structure used in our model. Section 2.2 and 2.3 describe how the lexicon and morphotactics are structured in our model. The rule representation for efficiently treating spelling changes is given in section 2.4. Section 3 is the detailed exposition of our predictive morphological analysis model. In section 3.1, it is presented how dynamic data structures for predictive analysis are represented and maintained in our model. Section 3.2 describes how the spelling rules in Stratum 1 are selected according to the input string without dictionary lookup, and how efficient segmentation is achieved through deterministic application of spelling rules and structure sharing. Section 3.3 deals with global morphotactic checking for possible morpheme sequences analyzed by the morpheme segmentation. Section 4 reports the empirical results of our implemented system. Section 5 summarizes the work.

2. The Representation of Morphological Knowledge

To process words computationally, four kinds of morphological knowledge
are required: (1) alphabet (2) lexicon (3) morphotactic descriptions and (4) morphophonological rules (Koskenniemi (1983), Ritchie et al. (1991), Sproat (1992)). This section discusses how the morphological knowledge is expressed in our model.

2.1 Alphabet

Words are represented as a sequence of letters in the alphabet of a particular language. In our model, a letter sequence for a word is also interpreted as the repetitive sequence of onset, peak and coda which we assume constitute a syllable. In this sense, words in our model can also be considered to be a sequence of syllables each of which has three letters belonging to onset, peak and coda in that order. The following is the set of Korean alphabetic letters classified in terms of the position they may take in the syllable template.¹

\[
\text{Onset} = \{0, k, kk (G), n, t, tt (D), l, m, p, pp (B), s, ss (S), c, cc (X), ch (C), kh (K), th (T), ph (P), h\}
\]

\[
\text{Peak} = \{0, a, ay (E), ya, yay (yE), e (6), ey (e), ye (y6), yey (ye), o, wa, way (wE), wi (8), yo, u, we (w6), wey (we), wui (v), yu, u (4), uy (4y), i\}
\]

¹ Throughout the paper, the Yale romanization is used to represent the Korean alphabet.
Coda = \{0, k, kk (G), ks, n, nc, nh, t, l, lk, lm, lp, ls, lth (lT), lph (lP), lh, m, p, ps, s, ss (S), ng (g), c, ch (C), kh (K), th (T), ph (P), h\}

0 is the orthographic "filler" letter, and parenthesized letters are our system codes used in the actual implementation of our model.

(a) The right-branching structure  
(b) The left-branching structure  
(c) The hybrid structure

Figure 1: The syllable structures.
In the literature, the syllable structure of Korean is assumed to have the form in Figure 1(a), in which peak (nucleus) and coda is grouped as rhyme. With this syllable template, the detailed schematization of syllable structure diverges, depending on whether a glide is grouped with onset or peak (Ahn (1986), Lee (1982), Kim-Renaud (1978)). Another possible syllable structure is the left-branching struture in which onset and peak is grouped as core, as in Figure 1(b). In this paper, we adopt a hybrid syllable structure integrating these two syllable structures, in which peak is shared by core and rhyme (see Figure 1(c)). The reason for this is practical rather than theoretical. Many of morphographemic changes in Korean (and possibly other languages) are governed by the core of a boundary syllable. But we cannot exclude the right-branching syllable structure because there are morphological phenomena that can be only described in terms of referring to the rhyme of a boundary syllable.

In addition to the basic letter set, we also make use of archiphonemic variables such as Vowel (representing the set of vowels) and Light (the set of light vowels = {a, o}) to make morphological rules more compact and readable (Koskenniemi (1983), Karttunen (1983)).

2.2 Lexicon

The lexicon in our model is represented in the form of letter trees (i.e., trie) known to be an efficient data structure for searching sets of letter strings (Knuth (1973)). In the trie structure for a set of letter strings, each path from the initial state to a final state corresponds to a word. Figure 2 depicts the trie lexicon representing the set of the following morphemes: to (toO)²
'degree', top (top) 'to help', towa (toOwaO) 'tile', wa (OwaO) 'along with', and e (06O) 'INFINITIVE'. In Figure 2, the circles are individual states; the lines (or arcs) connecting the states represent the transitions; the label on a line represents a character in the word that leads the transition from one state to another; the state $s_0$ is the initial state and the doubly-circled states are the final states; the final state indicating the end of the path for a word carries the detailed lexical information.

![Diagram](image)

**Figure 2. The trie lexicon**

2) "toO" is the internal representation of the open syllable "to" in which the coda position of the syllable is filled with the filler "O". This is because each position in a syllable is assigned a letter in our implementation even though it is phonologically empty.
2.3 Morphotactics

Being an agglutinative and highly inflected language, Korean uses affixation and compounding as the major mechanisms for word formation. These processes are active in both verbal and nominal morphology. The morphotactics of verbal and nominal words of Korean is expressed as a finite state transition network (FSTN) à la Koskenniemi (1983), as in Figure 3. This FSTN shows how morphemes can combine to form valid verbal and nominal words.

The morphotactics in our model is basically identical with Koskenniemi’s two-level model in that the compatibility of two adjacent morphemes is defined in terms of a finite state machine. However, while the morphotactic description of the two-level model is limited to the “local” connection of two morphemes, in our model the “global” connection of the morphemes in a complex word is also taken into consideration. In the two-level model, for example, the word *enrichment* is analyzed as en+rich+ment, satisfying the local morphotactic constraints that the prefix *en-* can combine with an adjective (i.e., *rich*) and adjectives combine with the noun-making suffix *-ment*. The problem with this kind of local morphotactic checking is that words like *richment* are also recognized to be legitimate because all the morphotactic constraints are defined between two adjacent morphemes (Karttunen and Wittenburg (1983), Sproat (1992)). In our model, the “global” morphotactic checking as well as “local” morphotactic checking is

3) “G” stands for the vowel *a* in our system.
4) In the two-level model, the restriction that excludes such as *richment* and *tailment* cannot be expressed without adjective and nouns duplicated in the lexicon, costing a great deal of memory resources let alone theoretical elegance.
adopted to avoid discontinuous dependencies of this kind without having two different dictionaries containing identical sets of words. In our model, we can prevent nonwords like *richment from being accepted by checking if the segmented input word conflicts with the global morphotactics encoded as a FSTN, as in figure 3.

![Diagram of morphological structure of Korean words]

*Figure 3: The global morphological structure of Korean words.*
2.4 Spelling Rules

In agglutinative languages like Korean, morphophonological rules are locally applied in the sense that they only involve two adjacent morphemes (Koskenniemi (1983), Ritchie et al. (1992)). That is, when two morphemes (the left morpheme and the right morpheme) are concatenated, the boundary spelling between the two morphemes is changed according to their morphophonological environment. Depending on which of two adjacent morphemes is changed, there are four possible types of morpheme concatenation:

- **Type 0** (No alternation). There is no spelling change in combining the left morpheme with the right morpheme. For example, the verb stem *swum* ‘to hide’ plus the final ending *e* ‘INFINITIVE’ forms the verbal word *swum-e* without any spelling change.

- **Type 1** (Left alternation). Only the left morpheme has some spelling changes. For example, a verb stem *tut* ‘to hear’ plus the final ending *e* ‘INFINITIVE’ forms the verbal word *tut-e*, replacing the final consonant *t* of the left morpheme by *l*.

- **Type 2** (Right alternation). Only the right morpheme has some spelling changes. For example, a verb stem *pat* ‘to receive’ plus the final ending *e* ‘INFINITIVE’ forms the verbal word *pat-a*, replacing the vowel *e* of the right morpheme by *a*.

- **Type 3** (Left and right alternation). Both the left and right morphemes have some spelling changes. For example, the verb stem *top* ‘to help’ plus the final ending *e* ‘INFINITIVE’ forms the verbal word *to-wa*, with spelling changes of both morphemes.
Figure 4 shows the local description of morpheme concatenation of two adjacent morphemes. The surface level represents the letter sequence of the input string, and the lexical level represents the morpheme sequence recovered from morphological analysis on the surface level. UCL is the unchanged portion of the left morpheme, and UCR, the unchanged portion of the right morpheme. UCL and UCR in the lexical level are directly mapped onto those in the surface level. CL in the lexical level corresponds to the changed portion in the left morpheme of the surface level, and CR, the changed portion in the right morpheme. The *alternation position* of the left morpheme is the starting position of spelling changes which depends on what its right morpheme is. If the left morpheme has no spelling change, the alternation position is the same as the end of the left morpheme. LAS
(left alternation string) indicates the lexical substring in CL. In the case of Type 0 and Type 2 where the left morpheme undergoes no spelling change, the length of LAS is zero. RAS (right alternation string) indicates the lexical substring in CR. In the case of Type 0 and Type 1 where the right morpheme undergoes no spelling change, the length of RAS is zero. Alternation length is the length of the alternation part in the surface level, which corresponds to the sum of length of LAS and RAS.

In this section, we briefly explain how some phonological phenomena of Korean are described in our rule formalism based on the syllable structure. Then, we discuss how the spelling rules are interpreted and represented.

2.4.1 The linguistic Representation

In theoretical phonology, it is common practice to describe ordering relations between phonological rules. In Korean, the $p$-irregularity and vowel harmony rule for dealing with verbal words such as $tōwa$ 'to help' (analyzed as the verb stem $top$ plus the infinitive suffix (ending $e$) would be a typical example that requires the application order between the two rules. The vowel harmony rule specifies the constraint that if the final vowel of a verb stem is $a$ or $o$, the ending $e$ must be changed into $a$. The $p$-irregularity rule specifies the constraint that if the final consonant $p$ of a verb stem, when it is followed by the vowel $a$ and is a single syllable, changes to the semivowel $w$, and then the semivowel $w$ and the vowel $a$ yield the compound vowel $wa$. As a result, the verb stem $top$ and the ending $e$ form the word $tōwa$ by two steps: first, the ending $e$ is changed into $a$ by the vowel harmony rule; then, $top$ plus $a$ becomes $tōwa$ by the $p$-irregularity
rule.

Since the order of "undoing" the spelling rules in morphological analysis is the reverse of applying them for word formation, the undoing of the p-irregularity rule precedes the undoing of the vowel harmony rule. In our model, this rule ordering is expressed in terms of two rule strata5.

- **stratum 1:** Most of the irregularity and contraction rules in Korean morphology belong to this group. In morphological analysis, the rules of Stratum 1 are responsible for discovering and recovering the left morpheme of inflected (Type 1 or Type 3 alternation) words.

- **stratum 2:** In Korean morphology, the vowel harmony rules and the u-deletion rules are typical ones in this group. These rules are responsible only for recovering the right morpheme after the left morpheme of the input word is segmented.

The order of undoing spelling rules exists only between the two strata: first, undo the rules in Stratum 1, then undo the rules in Stratum 2 with the result from applying the rules in Stratum 1. The rules in a stratum are independent of each other.

5) In the two-level model, there are no ordering relations between morphophonological rules. All the rules are applied at one fell swoop, whether they are applied in the serial or parallel manner. This does not mean, however, that the rules that require order specifications cannot be expressed in the two-level model, though it can be argued that it lacks descriptive adequacy. In our model, the adoption of rule ordering has also been motivated for computational reasons, i.e., for predictive applications of rules which enables us to avoid redundant computations. See section 3.
Using our rule formalism, the *p*-irregularity and vowel harmony rules can be written as follows:

- The *p*-irregularity:
  
  **R1.** \( wa(\text{core}, i) := p + a \Leftrightarrow \{a, o\}(\text{rhyme, i-1}) \& i = 2 \)
  
  / [left [cat=\text{verbal}] [irregular=p]]

Where (1) '\( wa(\text{core}, i) \)' indicates the core \( wa \) in the \( i \)-th syllable of an input word, (2) '\( p+a \)', the changed lexical letters around the morpheme boundary (i.e., \( \text{LAS}(p) \) and \( \text{RAS}(a) \)), (3) '\( \Leftrightarrow \)', the substitution of the corresponding lexical part (\( p+a \)) for the surface alternation part (\( wa \)), (4) '\( \Leftrightarrow \)', that the substitution occurs if and only if the phonological and morphological contexts are satisfied, (5) '\( \{a, o\}(\text{rhyme, i-1}) \)', the left (phonological) context of the substitution (i.e., the substitution occurs after which the rhyme of the (\( i-1 \))-st syllable is \( a \) or \( o \)), (6) '\&', an AND operation, (7) '\( i=2 \)', the syllable position constraint (i.e., the value of syllable position variable \( i \) is 2), (8) 'left', the left morpheme, (9) '\( \text{cat=\text{verbal}} \)', that the category of the left morpheme must be verbal, and (10) '\( \text{irregular=p} \)', that the left morpheme must have the *p*-irregularity.

- The vowel harmony:
  
  **R2.** \( a(\text{core, 1})(\text{morph, right}) := e \Leftrightarrow \{a, o\}(\text{peak, -1})(\text{morph, left}) \)
  
  / [left [cat=\text{verbal}]]

where (1) '\( a(\text{core, 1})(\text{morph, right}) \)' indicates the core \( a \) in the first syllable of the right morpheme, and (2) '\( \{a, o\}(\text{peak, -1})(\text{morph, left}) \)' denotes the
phonological context in which the rule can be applied only when the peak of the last syllable in the left morpheme is a or o.

R1 states: change the core \( \omega \alpha \) in the second syllable of a word into the consonant \( p \) to make a left morpheme and then insert the vowel \( \alpha \) just after the core if and only if the first syllable rhyme is \( a \) or \( o \), and the segmented left morpheme is verbal and has the \( p \)-irregularity. R2 states: replace the first core \( a \) of the right morpheme by \( e \) which is the basic form of \( a \) if and only if the final peak of the left morpheme is \( a \) or \( o \), and the left morpheme is verbal.

2.4.2 The Computational Representation

Figure 5(a) and (b) is the graphic representation of how spelling rules are interpreted according to their surface and lexical context. Whenever an input string contains "a00wa" or "o00wa" word-initially (specified by the syllable position constraint "i=2" in R1) as its substring, the analyzer takes these substrings to be a clue (i.e., the surface context) to the applicability of R1. Then, by consulting the segmentation information in the lexical level of R1 (i.e., "p+0a"), the analyzer looks up in the dictionary to get the lexical information of the possible morphemes. If the lexical information obtained from the dictionary lookup does not conflict with the one in R1("verbal" and "p-irregularity" in this case), R1 succeeds and the lexical substring is recovered according to the lexical level specified in R1.

Each of the spelling rules, regardless of the stratum it belongs to, is converted into two different executable data structures by the rule compiler.
(see Figure 6): "rule identifier" and "rule table".

The rule identifier, represented by a deterministic finite automaton (DFA), defines all occurrences of the surface context string of the rule. For example, the rule identifier for R1 in Stratum 1 can be represented as in Figure 6(a). In Figure 6(a), the numbered rows are individual states; final states are marked with a colon, and nonfinal states, with a period; state 0 is the initial state; numbers in the table represent state transitions; ' ' indicates an error; at the final state of the automaton, the rule number is stored to refer to the relevant rule table to be used for recovering the lexical string (see below). The automaton in Figure 6(a) accepts all the words partially having the string "a00wa" or "o00wa".

The rule identifier for R2 in Stratum 2 is composed of two DFAs, as in Figure 6(b): backward DFA for the left morpheme and forward DFA for the right morpheme. However, its function is the same as that of R1 except that it succeeds only if the final state of both DFAs is arrived at.

The rule table contains the information directly related to the rule undoing, i.e., restoration of the lexical string. The table is composed of 8 elements: (1) the rule number as a key of the table, (2) the alternation position, (3) LAS, (4) RAS, (5) the alternation length, (6) the final state constraint which specifies the substring position to which the predicted rule is relevant, (7) the category which is a morphological category the left morpheme can or cannot have, and (8) the irregular code which encodes the irregularity information of the left morpheme. Here the elements (6), (7), and (8) are to check if the undoing of each predictive rule is proper. The elements (3) and (6) are the information for undoing the spelling rules in Stratum 1 only.
Figure 5: The interpretation of R1 and R2
3. The Predictive Morphological Analysis

When an input word is ambiguous, the morphological analyzer should be able to produce all possible morpheme sequences for later disambiguation.
More importantly, the analyzer should be able to process such ambiguous words efficiently, because the ambiguity of a complex word (in Korean) can grow exponentially as the length of a word grows (See section 3.1.3). To achieve efficient processing of ambiguous words, two types of techniques are adopted in our predictive morphological analysis model.

- **Predictive Rule Application** to apply only the spelling rules suitable for a given word, not to apply all the spelling rules exhaustively.6
- **Dynamic Programming** to avoid redundant computations of analyzed substrings occurring when the input word is morphologically ambiguous.

This section describes how our model can efficiently deal with (complex and ambiguous) input words without doing redundant computations in rule application and morpheme segmentation (See Figure 7 for the overall control structure of our model). Roughly, morphological analysis in our model is done through two processes. One is the morpheme segmentation which decomposes an input word into its component morphemes. The other is the morphotactic checking which judges whether the segmentation is proper or not. The morpheme segmentation is based on the “binary” segmentation which divides the input string into two parts such that the left part is a morpheme, and the right part is the remaining substring in the input string. After the left morpheme of the input string is identified, the right substring is restored, and recursively processed by the same

---

6) In a morphological analyzer based on the two-level model, a complex word is segmented through blind applications of all the spelling rules and local morphotactic checking. As often pointed out in the literature (Barton et al. (1987), Sproat (1992)), this search strategy requires massive backtracking, causing fatal efficiency problems.
procedure. Global morphotactic checking for sequences of segmented morphemes is done after morpheme segmentation for a given input word is completed.

Figure 7: The overview of predictive morphological analysis
3.1 Dynamic Data Structures for Predictive Analysis

3.1.1 Rule Agenda

A rule agenda is a data structure that keeps the spelling rules predicted by the rule identifier for rule undoing. An item of the rule agenda consists of two elements: the rule number of the selected rule to refer to the rule table and the alternation position.

3.1.2. Substring Agenda

A substring agenda is a mechanism to keep the information record on substrings the analyzer should process. By using the substring agenda, the analyzer can find all possible morpheme sequences for an (ambiguous) input word. An item in a substring agenda consists of three elements, i.e., (1) a letter sequence $S$ for the current substring, (2) a sequence of integers $P$ indicating the position of the substring letters, and (3) a sequence of letters $L$ recovered by the morphological rule.

Let us consider the verbal word $kasita$ in (1) to demonstrate how the process of this three-way ambiguous word is viewed in terms of the substring agenda.

(1) a. $ka$ ‘to go’ + $usi$ ‘HONORIFIC’ + $ta$ ‘ENDING’.
   b. $kal$ ‘to grind’ + $usi$ ‘HONORIFIC’ + $ta$ ‘ENDING’.
   c. $kasi$ ‘thorn’ + $ta$ ‘ENDING’.
To begin with, the substring agenda only contains the input string and its position number, as in (2). The # symbols surrounding the input string indicate the word boundary.

(2) S:  \# k a s i t a \#
     P:  0 1 2 3 4 5 6 7
     L: 

After processing the substring “ka”, the analyzer will have two ways to proceed. The first option is to take the substring “ka” as a component morpheme and continues to work on the remaining substring “sita”. The second option is to hypothesize that some substring following “ka” constitutes another component morpheme (in this case, “kasi”). These two possible analyses shown in items (3) and (4) are appended to the substring table.

(3) S:  u s i t a \#
     P:  3 3 4 5 6 7
     L:  u

(4) S:  t a \#
     P:  5 6 7
     L: 

---

7) For the sake of exposition, we assume that the analyzer applied the spelling rule in stratum 2 (i.e., u-deletion), recovering the letter u before the substring sita. Notice that both u and s are assigned the position number 3. This is because s needs to keep the position information for later stage of processing.
Figure 8: The substring agenda for analyzing the Korean word *kasita*.

Figure 8 shows the full-fledged substring agenda for the word "kasita" in which the ambiguity of the word has been subsumed.

### 3.1.3 Dynamic Chart

The "dynamic" chart is a data structure which enables the analyzer to keep the partial results for efficient processing of ambiguous words. The dynamic chart is represented by a labelled directed acyclic graph (M, C, \( w_0, w_1 \)) where
M is a finite set of vertices. A vertex is labelled with the lexical information of a morpheme LEX, the starting position p of the next

Figure 9: Representing substring-shared structures in a dynamic chart.

8) The term "dynamic" is used in the meaning that since unlike syntactic parsing, the number of the component morphemes of an input word is initially unknown to the analyzer, each node is determined only after morphemes are identified by segmentation.
input substring and a sequence of letters $L$ recovered by the morphological rule in the next input substring. For the initial and final vertices, $LEX$ is the word boundary, $L$ is null, and $p$ is 0 and the position after the end of the input word respectively.

- $C$ is a set of edges between vertices. An edge directed from $m_t$ to $m_j$ ($m_t, m_j$) is labelled with a binary value (0 or 1). 0 means $m_j$ is not compatible with $m_t$ and 1 means $m_j$ is compatible with $m_t$.
- $w_0$ is the initial vertex in M, which indicates the left word boundary.
- $w_1$ is the final vertex in M, which indicates the right word boundary.

Let us take the verbal word $kasita$ as an example to see how the ambiguity is efficiently recorded in our dynamic chart. First, by traversing the substring agenda for $kasita$ in Figure 8, the analyzer will produce the chart, i.e., the labelled DAG, containing the information on the segmented word “ka+usi+ta”. Then, the analyzer goes on to explore the next possible segmented word to be “kal+usi+ta”. The component morphemes +usi and +ta in kal+usi+ta, however, are already in the chart, because the analyzer has stored them in the same string positions while processing “ka+usi+ta”. Thus, these two duplicate morphemes are not processed and the verb stem kal shares them with the previously analyzed verb stem ka, as shown in Figure 9. Though ka and kal shares usi in the chart, the compatibility of these two verb stems with the prefinal ending is independently checked, labelling the edges either 0 or 1. Since both ka and kal is compatible with the prefinal usi, the edges $(m_4, m_5)$ and $(m_7, m_2)$ are labelled “1” in the chart.

---

9) The traversal of a substring agenda works in the left-to-right depth-first manner in our implementation.
3.2 Predictive Morpheme Segmentation

3.2.1 Stratum 1 Rule Selection and Left Morpheme Segmentation

Given an input substring, the analyzer scans the input substring (i.e., $S$ in the substring agenda item) according to the rule identifier of Stratum 1.\textsuperscript{11} Whenever an input character arrives at a final state, the analyzer saves the rule table referred to by the final state in the rule agenda. If there is no rule in the rule agenda, we consider the input substring has no spelling change. In case there is no spelling change in the input substring, segmentation is done simply by dictionary lookup. Otherwise, each rule in the rule agenda is applied to the input substring for segmentation along with dictionary lookup:

1. Match the surface input substring with possible lexical entries from the leftmost letter through the position just before the “predicted” alternation position in the rule table. (A morpheme may be found through this step, in case there is no spelling change in the input substring.)

2. Match the UCL plus the predicted left alternation string (LAS) with possible lexical entries. If this fails, then ignore the rule. Otherwise, check if the searched morpheme conflicts with the lexical information in the rule table (e.g., category and irregularity); if not, then store the

\textsuperscript{11} Notice again that the process is carried out without dictionary search.
morpheme in the chart, else ignore the rule. Then, make a new input string (i.e., right morpheme) by concatenating RAS to UCR.

The morpheme identified this way is checked if it is compatible with the previously segmented morpheme (i.e., the left morpheme) by way of the local morphotactic information as discussed before. If the currently identified morpheme already exists in the chart (i.e., the morpheme has been processed before at the same position of the input substring recovered by the same spelling rule), it is not computed and shared (no dictionary lookup and morphological checking).

3.2.2 Stratum 2 Rule Selection and Right Substring Restoration

The intermediate output string is scanned again by the rule identifier of Stratum 2 (i.e., the backward DFA and the forward DFA). When the rule identifier arrives at a final state, the analyzer checks if lexical features in the previously segmented left morpheme are unified with the morphological context of the rule table referred to by the final state. If unified, the analyzer selects the rule. Then, the recognized substring is recovered in its lexical form by consulting the morphological information in the rule table. The information on the analyzed substring is stored in the chart and used as the predictive information on a possible sequence of morphemes for the input substring yet to be processed.
3.3 Global Morphotactic Checking

As noticed before, all the possible morpheme sequences are produced in our model, in case an input word is ambiguous. The ill-formed analyses among them are filtered out by the global morphotactic checking which can be described as follows:

- Check if each of the paths from the initial vertex \( w_0 \) to each of the final vertex \( w_1 \) satisfies the morphotactic constraints (i.e., paths in FSTN). If it does, make it as a valid output if the constraints are satisfied, otherwise discount it as an incorrect analysis.

4. Implementation and Empirical Results

The predictive morphological analyzer discussed in this paper has been fully implemented in C under the UNIX environment on SUN SPARC 10. We have constructed the dictionary that contains about 60,000 entries and 48 spelling rules (41 for Stratum 1 and 7 for Stratum 2) which we believe exhaust Korean verbal morphological phenomena.

The system has been tested on 413,975 words selected from various Korean elementary textbooks. For this test data, 95% of the input words were recognized. 10% of the analyzed words contained implausible results along with the correct analysis. This overgeneration is largely due to the weak morphotactic information based on only 13 categories. We believe
that this overgeneration problem can be overcome by dividing the existing categories into the more specified subcategories. Our system has failed to analyze 5% of the test data, due to unknown words and the lack of global morphotactic information.

To see the effectiveness of the predictive rule application method and dynamic programming, we have compared our system with a two-level morphological analyzer (Kim et al. (1994) with the morphological knowledge having the same coverage as our predictive system. While the two-level analyzer processed 5 words per second (200 msec per word in average), our predictive analyzer processed 154 words per second (6.5 msec per word in average), showing a considerable difference in performance. This test result show that the selective rule application and structure sharing techniques adopted in our system has a great effect on efficient processing of input words. In fact, we have learned from our experiment that 90% of the input data were processed predictively invoking less than 7 rules and almost all the input words were covered with 11 rules. This shows that our system apply the spelling rules only needed to analyze the input word, thus avoiding futile computations. Another point to note is that while the two-level morphological analyzer makes use of finite state transducers to map surface letters to lexical ones which lead to the serious problem of nondeterminism and backtracking (Barton et al. (1987)), our system deterministically scans the input letters through a finite state automaton which is used as a predictive rule identifier, not a letter-mapping transducer. The test results are summarized in the graph in Figure 10.
Figure 10: The average number of predictive rules related to a word.

This graph plots the number of words according to the number of spelling rules related to the words.

5. Conclusion

In this paper, we have presented a morphological analysis model of Korean that guarantees efficient and reliable processing. In so doing, we have introduced three types of new techniques. First, by adopting a new method
to interpret spelling rules, our model is able to do morpheme segmentation deterministically. Second, rule predictions guided by the rule identifier helps the analyzer to apply the spelling rules suitable to the input word. Third, the use of dynamic programming enables the analyzer to avoid recomputing analyzed substring when the input word is ambiguous. Our model has been experimented with 413,975 words randomly selected from the corpus of Korean elementary textbooks. Experimental results show that our model guarantees fast and reliable processing.
References


the English Lexicon, MIT Press, Cambridge, Massachusetts.
본 논문은 단어를 구성하는 모든 가능한 형태소열을 생성하는 효율적인 한국어 형태소 분석 모델을 제시한다. 본 논문의 형태소 분석 모델은 결정적인(deterministic) 철저규칙의 적용을 보장하며, 복합어나 중의성을 지니는 단어의 경우에도 불필요한 계산을 방지한다. 이러한 효율성의 획득은 (1)철저규칙을 해석하는 새로운 방법, (2)입력단어에 적합한 철저규칙만을 적용하는 예측중심의 규칙적용방법, (3)중의성이 있는 단어의 경우 이미 분석된 형태부분의 반복계산을 방지하는 동적 프로그래밍 기법의 사용에 의한 새로운 분석기술에 의하여 이루어진다. 본 논문에서 제시된 형태소 분석 모델은 국민학교 국어교과서에서 무작위로 추출된 413,975개의 단어를 대상으로 실험되었으며, 실험 결과는 본 모델이 효율적이면서도 견고한 형태소 분석을 보장하는 것으로 나타났다.

인공지능연구센터
전산학과
한국과학기술원
대전 305-701, 대한민국.