Incremental Construction of a Lexical Transducer for Korean[1]

Hyuk-Chul Kwon*, Lauri Karttunen
Dept. of Computer Science, Pusan National Univ. Pusan, 609-735, South Korea*
Xerox PARC, 3333 Coyote Hill Road,Palo Alto CA94304

ABSTRACT

The paper describes the construction of a lexical transducer for Korean that can be used for stemming and generation. The method contains two innovations: (1) two-level rules as well-formedness constraints in the initial phase; (2) the combination of intersection and composition of rule transducers in a deep cascade for the final result.

Keywords

Korean Lexical Transducer, Two-level Morphology, Morphotactics, Ordered Rules

1 Introduction

This paper presents an incremental construction method of a **lexical transducer** (LT) for Korean. A lexical transducer, first described by Karttunen, Kaplan, and Zaenen (KKZ) (Karttunen, 1992a), is a specialized finite-state transducer (FST) that maps canonical citation forms of words and morphological categories to inflected surface forms, and vice versa. LT's have many advantages for stemming, morphological analysis, and generation. They are

(i) **bidirectional:** the same structure can be used for stemming and generation.

(ii) **efficient:** the recognition and generation of word forms does not require the application of any morphological rules at runtime.

LT's for English and French have been built at *Xerox PARC* within a framework known as **two-level morphology** (Koskenniemi, 1983). As described by KKZ(Karttunen, 1992a), this can be done in three steps: (i) we construct a simple finite-state automaton (LA) that defines all valid lexical forms (LFs) of the language. A LF is a concatenation of stems and morphemes in their canonical dictionary representation. (ii) We describe morphological alternations by means of **two-level rules**(Koskenniemi, 1983; Karttunen, 1993), compile the rules to finite-state transducers, and intersect them to form a single rule transducer (RT). (iii) We merge the LA with the RT by composition producing the LT that has on its lexical side every valid lexical form of the language and on the surface side the corresponding realization as determined by the morphological alternations of the language.

KKZ argued that for French, it was best to divide step (ii) into two stages. A three-level description was required to give a linguistically satisfactory account of the plural formation of compound nouns. KKZ opted for two cascading two-level rule systems that are compiled separately, then intersected laterally and finally composed to a single RT.

The task of building a morphological analyzer for a language such as Korean or Japanese is a much higher challenge than it is for English and French. A Korean verb may have more than fifty thousand inflected forms.[2] The Korean writing system (*Hangul*) does not consistently distinguish between single and compound nouns. Because *Hangul* uses syllabic characters, changes in syllable structure are directly reflected in the orthography.

Because of the complexity of the morphological alternations in Korean, it is very difficult, although not impossible in principle, to describe them in a single two-level rule system or in a system that is limited to just three levels like the KKZ system for French. The most natural description of the Korean alternation is a cascade of rules of greater depth.

2 Morphological Alternations in Korean

The *Hangul* is a phonemic syllable-based script where morphological alternations that change the syllable structure of the word are reflected in the orthography (Korean Ministry of Education, 1988; Kim, 1990). This paper uses the so-called Yale system for representing Hangul in a Romanized form, except that we

[1] This paper was partially supported by Korean Science and Engineering Foundation.

[2] A "*e-jel*(word)" which is a spacing unit of *Hangul* can consist of a verb stem, several endings and postpositions. The A.I Lab of Dept. of Computer Science, Pusan National Univ. has more than 50,000 "*e-jel*" generated from "*mek-ta*(eat)"

use *wue* and *oa* instead of *we* and *wa* of the Yale system because *we* and *wa* do not show that they are diphthongs, composed of *wu* and *e* and of *o* and *a* respectively.

Examples (1) and (2) involve three simple morphological alternations: (i) the realization of a stem final *p* in irregular predicates as a vowel in front of vowel-initial suffixes; (ii) left-to-right vowel harmony based on partitioning of vowels into 'light' ([+light]:*a*, *o*, *oa*), 'dark' and 'neutral'([-light]); (iii) the realization of a morpheme boundary as a syllable boundary or as nothing.

A syllable boundary is introduced before the last consonant of irregular -*p* verbs/adjectives when a vowel-initial suffix follows and the -*p* itself is realized as *o* if the preceding vowel is [+light], otherwise *wu* by vowel harmony. Only some of the predicates ending in -*p* are irregular. In verbs that end in a vowel such as *cwu* 'to give', the vowel may merge with a suffix-initial vowel to form a diphthong or it may retain its syllabic status in a two-vowel sequence.

We use "+" in the lexical representation to mark morpheme boundaries, "-" to mark syllable boundaries, "0" to represent deletion (surface side) and epenthesis (lexical side), and two diacritic markers {*p*Verb} for an irregular -*p* verb and {*r*Verb} for a regular verb to represent classes of verbal stems.

```
(1) (a) c wu 0 p   {pVerb} + a/e - s e
        (cwup{pVerb}+a/e-se)
    (b) c wu - wu  0          0  e  - s e
        (cwu-wue-se: to pick up)
```

```
(2) (a) c wu {rVerb} + a/e - s e (cwu{rVerb}+a/e-se)
    (b) c wu   0   -   e   - s e (cwu-e-se: to give)
    (c) c wu   0   0   e   - s e (cwue-se: to give)
```

The (a) part of both (1) and (2) are lexical forms and (b) and (c) are corresponding surface words.

Because *cwup* is an irregular -*p* verb, the following phoneme *a/e* is a vowel and the preceding syllable *wu* is [-light], *p* in (1) (a) is realized as *wu*. The *a/e* is realized as *e* because the preceding surface vowel *wu* is [-light]. At the same time, *wu* and *e* are contracted into a diphthong *wue* which is described as the deletion of "+" in (a) of (1). These two changes are linked in that one must not be allowed to happen without the other. Otherwise *cwu-wu-e-se* and *cwu-wue-se* would be generated, but only *cwu-wue-se* is grammatical. On the other hand, in the case of the regular verb *cwu*, both *cwu-e-se* and the contracted variant *cwue-se* are acceptable.

These rules can be described easily by two-level morphology as follows.

(3) (i) A syllable boundary ("-") is introduced before a stem-final *p* in irregular -*p* verbs/adjectives when a vowel-initial suffix follows.

(ii) A stem final *p* in irregular -*p* verbs/adjectives is realized as *o* if the preceding vowel is [+light], otherwise *wu*.

(iii) *a/e* is realized as *a* if the preceding vowel is [+light], otherwise *e*.

(iv) (a) The morpheme boundary following irregular -*p* verbs/adjectives is deleted before a vowel-initial suffix and realized as syllable boundary elsewhere.

(b) The morpheme boundary in regular verbs/adjectives can be deleted or realized as a syllable boundary depending on context.

With the help of the Xerox two-level rule compiler ('twolc')(Karttunen, 1992b) the rules can be compiled to finite state transducers and intersected to a single transducer. Describing such phenomena as parallel rules may be complicated because each rule may be a formulation of effects caused by several phonological rules. For example, in formalizing (ii) as a two-level rule we must take into account both irregular conjugation of -*p* verbs/adjectives and vowel harmony. This is a not a desirable state of affairs. We will come back to this point later.

3 Construction of a Korean Lexical Transducer(LT)

The first step in the construction of a lexical transducer is to create a simple finite-state automaton for all valid lexical forms of Korean. The lexical automaton (LA) is composed with the first set of rule transducers (RT). The resulting transducer has on its "upper" side, the valid lexical forms, and on the "lower" side, intermediate representations derived by the first set of rules. This intermediate transducer is composed with the second set of rule transducers and the process is iterated several times. At each stage in the process, the lexical side remains unchanged and the intermediate representations are changed by the new set of rules. The final result is a transducer that associates the valid lexical forms with their proper surface realizations. Conceptually this is similar to what happens in a traditional phonological derivation. However, note that rules apply to the lexicon as a whole rather than to individual words and the result of each application is a new transducer. Because the intermediate levels disappear in the composition, the resulting LT is equally well suited for morphological analysis as it is for generation.

The compilation and intersection of rule transducers was done with the twolc compiler, the construction of the LA and the compositions we carried out with the Xerox interactive finite-state calculus ('ifsm').

3.1 Construction of Lexical Automaton(LA)

The ifsm-utility enabled us to assemble the LA incrementally. The first step was to divide the total list of morphemes into sublexicons on the basis of their morphological type and to make a text file for each sublexicon. We added diacritic markers to the edges of certain types of morphemes in order to be able to enforce morphotactic constraints on valid morpheme sequences.

Each sublexicon was compiled separately to a finite-state automaton. The sublexicons were used to construct the LA with the help of the regular expression facility in the ifsm-toolkit. For example, having compiled a simple automaton from the list of simple nouns, we could expand it to an infinite lexicon of compound nouns with the regular expression

"noun.auto" [# "noun.auto"]*

This regular expression reads the noun automaton from a file and concatenates it with itself any number of times and marks the internal word boundaries with #.

The first version of the LA was made in this way by combining sublexicons with regular operations (concatenation, union, iteration).

In order to enforce morphotactic constraints on the concatenation of some classes of suffixes, we wrote a set of two-level rules that require or prohibit the occurrence of particular diacritics at certain suffix boundaries. Lexical forms that do not satisfy the morphotactic constraints get eliminated in the composition with the well-formedness rules. The diacritics themselves are realized as zero so that they are not present in the lower side of the resulting transducer. The final form of the lexical automaton is obtained by extracting the lower-side from that transducer as a simple automaton.

We believe that this incremental method of lexicon construction is better suited to morphologically complex languages than the lexicon format commonly used in two-level morphology. In standard two-level lexicons, individual entries contain information about which sublexicon they may concatenate with. The entire lexical structure is compiled in one step to large letter tree (Karttunen, 1993; Antworth, 1990). Our method is more tractable in two ways. Firstly, the lexicon can be developed and refined stepwise. Secondly, the morphotactic rules of the language are described explicitly as the regular expressions that construct the LA in conjunction with the well-formedness constraints that eliminate certain types of concatenations. In two-level lexicons of the standard variety, the morphotactic structure of the language is not described explicitly at all. Rather, it is expressed in a very opaque and indirect way, in the sequences of links between entries and sublexicons.

Sproat argued that two-level morphology of morphotactics leads to a somewhat inelegant model of long-distance dependencies and suggested the unfication scheme, due to Bear, as a solution (Sproat, 1992). But unification scheme introduces additional runtime overhead. The above approach can easily and explicitly describe the fact that "-able" attaches to verbs formed with the prefix "en-" and does not require additional runtime overhead.

We give a few examples of the difficulties in the description of Korean morphotactics. There are two different types of endings: (i) non-final (verbal) endings for tense, modality, subject honorific or aspect, and (ii) final (verbal) endings as complementizer, nominalizer and adjectivizer. The non-final endings are placed in front of final endings and must be followed by a suffix of the second type.

(4) shows the ordering restrictions of non-final endings. The parentheses indicate optionality.

(4) (+ Hon) (+ Past + Perf (+ Will)

| (+ Past) (+ Will) (+ Retro))

(Hon:Honorific; Retro:Retrospection;

Perf: Perfect Aspect)

(4) compiles to a lexicon covering 20 different compound non-final ending sequences including null. This representation is clearly more informative than a simple listing of the members of the class. The prohibition of "Past+Perf+Will+Retro" in (4) can not be described by an adjacency table.

In (4) we do not need any morphotactic diacritics on the left because all non-final endings can combine with any verb and adjective stems and the combination of non-final and final endings is controlled by the diacritics of the latter group.

(5) shows three entries in the sublexicon of final endings. The elements in square brackets are morphotactic diacritics. (Square brackets indicate grouping, the vertical bar marks a disjunction.) The diacritics are deleted by well-formedness rules when the final endings are combined with other morphemes. The diacritics on the left of *nun* and *nun-ka* shows that they can not combine with adjectives.

(5) [Verb | Adj | Hon | Past | Will | Perf] *t a* {Dec} ;

[Verb | Hon] + *n u n* {Con} ;

[Verb | Hon | Past | Will] + *n u n* - *k a* {Que} ;

({Con}:Conjunction; {Dec}:Declarative; {Que}:Question marking; ";": the end of declaration, the meaning is the same as "|")

The diacritic markers {Dec}, {Que} and {Con} have two roles as the feature of the morphemes and as the right-hand context. They remain in final LA because they are the feature of each morpheme.

By concatenating the subnetworks of compound non-final endings and final endings, we get a sublexicon of ending sequence as shown in (6). The [Verb | Adj] diacritics indicate that non-final endings can combine with any verb stems and adjective stems.

(6) ([Verb|Adj] "compound-non-final-ending.auto" +)
"final-ending.auto"

This concatenation produces an initial lexicon of 97498 (2*20*2378+2378) different sequences where 20 is the number of compound non-final endings and 2378 is the number of sequences of final endings with their morphotactic diacritics. This set is reduced to 7888 by a set of well-formedness rules that eliminate unwanted sequences and delete the morphotactic diacritics. The composition of the initial lexicon with the well-formedness rules produces a transducer from which the lower side is extracted as a simple automaton and used in the construction of the final LA.

Allowing nouns to freely compound with nouns creates a problem because it gives rise to many unacceptable or unlikely compounds. For example, the form *cwung-kan-i* has five alternate analyses:

(7) *cwung-kan-i*

(a) *cwung-kan*(middle)+*i*(subject marker)

*(b) *cwung-kan*(middle)#*i*(louse)

*(c) *cwung*(monk)#*kan*(liver,saltiness)
+*i*(subject marker)

*(d) *cwung*(monk)#*kan*(liver,saltiness)#*i*(tooth,louse)

*(e) *cwung*(monk)#*kan-i*(handiness)

Our solution was to constrain compounding with a well-formedness rule that excludes compounds with monosyllabic nouns (Kwon, 1990). The complexity of the morphological alternations in Korean is so high that we need an easy way to give constraints incrementally. Our approach is a consistent and explicit way of describing morphotactic rules including long-distance dependencies.

3.2 Composition of Lexical Automaton with Rule Transducers

After constructing the Korean LA, we derive from it a lexical transducer by composing the LA with rule transducers (RTs) in several stages. At each stage the previous result is composed with an RT derived by intersection from several two-level rules. The rule sets include (i) morpheme generation rules, (ii) rules for irregular verbs/adjectives, (iii) deletion rules, (iv) vowel harmony rules and (v) contraction rules. Morpheme generation rules give a surface realization to morphological tags, such as Past, Hon(orific), etc. Rules for irregular verbs deal with final consonants and syllabification. Deletion rules eliminate one of two adjacent vowels on morpheme boundaries. Vowel harmony rules realize the harmonizing archiphonemes *WU* as *o* or *wu* and *E* as *a* or *e* depending on the quality of the preceding vowel. Contraction rules involve the merging of adjacent vowels to a diphthong or a single vowel as a result of the loss of the intervening syllable boundary.

Although it is possible in principle to write just one two-level rule system that describes all the alternations in parallel, it is very difficult in practice to create a rule system with that degree of complexity. The complexity arises from the fact that the formulation of every rule in a two-level system depends on every rule that has some effect on the context of the rule that we are trying to express. For example, if there is a rule that forces X to be deleted in front of a Y and another rule that introduces Z between X and Y, great care must be exercised by the rule writer to make sure that both rules are specified in a way that leaves room for the other rule to have its effect but does not depend on it if the deletion of X and the insertion of Z are two independent alternations. The partioning of rules into sets and the interleaving of intersection and composition greatly simplifies the task of creating and updating the rule system. Rules that apply in different environments and do not affect each other can be compiled and intersected easily, whereas rules that involve alternations in overlapping contexts are most easily handled by placing them in different levels in the cascade. In effect the rules are partially ordered. Sproat also noticed that rule interactions which may be easy to state in terms of ordered rules, are often much more difficult to state in one two level rule system (Sproat, 1992).

For Korean, the partitioning of the rules for morphological alternations into the five sets described above appears to be the optimal choice. Each of the rules in the *Hangul* standard orthography published in March of 1988(Korean Ministry of Education, 1988) is described in the corresponding two-level rule separately in our implementation. The order of rules takes the role of rule interactions. In this cascade, the alternations described in section 2 as example (3) are split between three levels:

(8) (i) Rules for irregular predicates:
A syllable boundary is introduced before the stem-final *p* in irregular -*p* verbs/adjectives when a vowel-initial suffix follows. The following morpheme boundary is deleted and *p* is realized as the harmonizing archiphoneme *WU*

(ii) Vowel harmony rules:

(a) The *WU* is realized as *o* if the vowel of the preceding surface syllable is [+light], otherwise *wu*

(b) The *E* is realized as *a* if the vowel of the preceding surface syllable is [+light], otherwise *e*.

(iii) Contraction rules:
The morpheme boundary "+" can optionally be deleted between *wu* and *e*.

The effect of these rules with respect to the irregular -*p* verb *cwup* 'to pick up' is shown in (9).

```
(9) (a) c wu 0   p   {pVerb} + E - se
    (b) c wu - WU        0     0 E - se
    (c) c wu -   wu      0     0 e - se
```

The intermediate level, (b), is eliminated in the cascade, thus the final lexical transducer maps (a) directly to (c).

4 Conclusion

The success of our work on Korean further underscores the point made by KKZ(Karttunen, 1992a) that the most salient property of two-level morphology is not the number of levels but the fact two-level rules describe regular relations (just like classical phonological rewrite rules) (Kaplan, 1988; Ritchie, 1992). Consequently, it is possible to combine sets of parallel two-level rules by intersection and merge them with the lexicon and other rule systems in a cascade of compositions. The complexities of Korean morphology make it desirable both for linguistic and computational reasons to allow for many more intermediate levels than assumed in previous works on English and French. Regardless of the number of intervening levels, the outcome is a single lexical transducer that directly maps lexical forms to their inflected surface realizations, and vice versa.

In the construction of the lexical automaton for Korean, we have put two-level rules to a novel use as well-formedness constraints on lexical forms. The sublexicons from which the LA is constructed contain diacritic marks on the outer edge that identify the type of morphological constituents that the lexicon contains. The role of rules in the LA constructions is to enforce morphotactics and, at the same time, to eliminate the diacritics that encode them.

Theoretically, we can get the same LT to compose the morphotactic and phonological rules all together into one rule and compose it with the initial LA or to compose the initial LA with each rule of the morphotactic and phonological rules one by one in order. Practically, the composition of all the morphotactic and phonological rules into one rule causes the combinatorial explosion of states. This shows that ordered rules can be used to avoid the combinatorial explosion of states in one two level rule system too.

References

[1] Antworth, Evan L.(1990) *PC-KIMMO: a two-level processor for morphological analysis.* Occasional Publications in Academic Computing, No. 16,Summer Institute of Linguistics, Dallas, Texas.1990

[2] Kaplan, R. M.(1988) *"Regular models of phonological rule systems".* Alvey Workshop on Parsing and Pattern Recognition. Oxford University, April, 1988

[3] Korean Ministry of Education.(1988) *Hangul Standard Orthography (Revised in 1988)*, Document number 88-1, Published in March 1988.

[4] Karttunen, Lauri, Kaplan, Ronald M., and Zaenen, Annie.(1992a) *"Two-Level Morphology with Composition".* Coling-92. Proceedings of the fifteenth International Conference on Computational Linguistics. Volume I. pp.141-148. 1992.

[5] Karttunen,
Lauri and Beesley, Kenneth R.(1992b) *Two-Level Rule Compiler.* Technical Report. Xerox Palo Alto Research Center. ISTL-92-2. October 1992. [P92-000149]. Palo Alto, California. 1992.

[6] Karttunen, Lauri.(1993) *"Finite-State Constraints".* To appear in The Last Phonological Rule, John Goldsmith, ed. Chicago University Press. Chicago. 1993.

[7] Kim, C.(1990) *The Explanation of New Hangul Standard Orthography,* Kul-Sup Press. Seoul. 1990.

[8] Koskenniemi, K.(1983) *Two-level Morphology. A General Computational Model for Word-Form Recognition and Production.* Department of General Linguistics. University of Helsinki. 1983.

[9] Kwon, H. and Chae, Y.(1991) *"A Dictionary-Based Morphological Analysis".* Proceedings of the Natural Language Processing: Pacific Rim Symposium'91. pp.178-185, 1991.

[10] Ritchie, Graeme D.(1992) *"Languages Generated by Two-level Morphological Rules".* Computational Linguistics, No. 18, Volume. 1, pp.41-59. March 1992.

[11] Sproat, R.(1992) *Morphology and Computation,* MIT press, 1992.