PC-KIMMO-based Description of Mongolian Morphology

Purev Jaimai*, Tsolmon Zundui*, Altangerel Chagnaa, and Cheol-Young Ock****

Abstract: This paper presents the development of a morphological processor for the Mongolian language, based on the two-level morphological model which was introduced by Koskenniemi. The aim of the study is to provide Mongolian syntactic parsers with more effective information on word structure of Mongolian words.
First hand written rules that are the core of this model are compiled into finite-state transducers by a rule tool. Output of the compiler was edited to clarity by hand whenever necessary.
The rules file and lexicon presented in the paper describe the morphology of Mongolian nouns, adjectives and verbs. Although the rules illustrated are not sufficient for accounting all the processes of Mongolian lexical phonology, other necessary rules can be easily added when new words are supplemented to the lexicon file.
The theoretical consideration of the paper is concluded in representation of the morphological phenomena of Mongolian by the general, language-independent framework of the two-level morphological model.

Keywords: natural language processing, two-level morphological rule, Mongolian morphology, finit e-state transducers, computational linguistics

1. Introduction

This paper presents a two-level morphological description for the Mongolian language. A formal background, concerning the method of morphological analysis based on the two-level morphology, was stated by Koskenniemi [7]. Some aspects of Mongolian morphology are discussed in Section 3. The development of Mongolian morphology has been described in Section 4, by terms PC-KIMMO (Version 2.1) package as the implementation tool for two-level morphology.

2. Theoretical Background

The two-level morphology is a general, language-independent model, which is considered state-of-the-art in computational morphology by many researchers [7;10] and has been implemented for several different languages such as Finnish [7], English [11], Turkish [8], Japanese [1], etc.

Two types of representations are considered in the two-level morphological model: the surface represen-tation which would be the phonemic form of the word and the lexical representation which can be considered as the internal structure of the word, constructed by the concatenation of appropriate morphemes. The two-level rules can be transformed into Finite-State Transducers (FST) either by hand [2;7] or by using a special tool, which automates the process [5;6]. FST directly relate the two representations to each other.

Briefly, the first LISP implementation of Koskenniemi's two-level processor is developed by Karttunen [5]. But, the most well known version of the processor is the PC-KIMMO developed by Antworth [2] at the Summer Institute of Linguistics (SIL).

The description of Mongolian morphology is determined by Koskenniemi's two-level approach. The aim of the study is to provide Mongolian syntactic parsers via more effective information on word structure of Mongolian words. The rules for Mongolian are first written by hand and then compiled by a rule tool for PC-KIMMO into "state transition tables" in order that they can work in the finite state automation (FSA). It is the most fundamental mathematical model in language processing and designed for processing a regular language well-formed string. The work was initially done by a two-level rule compiling tool called "Kgen", and programmed by Nathan Miles in 1991. The current version number of Kgen is 0.2.

Fig. 1 shows an example of the rule and its compiled state table was created for the Mongolian morphological analysis.

The role and meaning of this rule was described in the section as "Rules Component". There are three states and six correspondences including default <@:@>. State 1 was marked the only final state with a colon and the others

Manuscript received September 29, 2005; accepted November 14, 2005. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).
Corresponding author: Cheol-Young Ock
* School of Information Technology, National University of Mongolia, Mongolia ({purev, tsolmon_z}@num.edu.mn)
** School of Computer Engineering and Information Technology, University of Ulsan, Korea (goldenl, okcy}@mail.ulsan.ac.kr)

non-final states marked with a dot symbol. The operator **(**=>**)** indicates an optional environment.

The reader who is interested in extensive reference to the two-level model introduced in [2;4;5;7].

```
'^' Sign Changing Rule
^:i => ___+:0 [V:0|MG|h]
          (a)

      ^  +  V  MG  h  @
      i  0  0  MG  h  @
 1:   2  1  1   1  1  1
 2.   0  3  0   0  0  0
 3.   0  0  1   1  1  0
          (b)
```

Fig. 1. (a**)** Sign Changing Rulem **(**b**)** State table relevant to this rule

3. Aspects of Mongolian Morphology

Basically, Mongolian is an agglutinative language in terms of structure, and most closely related to Turkish and Korean. Nowadays, the Mongolian language uses two official scripts: The **(**new**)** Cyrillic Mongolian script, the **(**old**)** Mongolian script. But, the first one is predominantly used. Thus, it was used in this work. Mongolian morphology has been described in this paper as mainly based on [3;9].

The **(**new**)** Cyrillic Mongolian alphabet consists of 35 letters. There are 13 vowels: *а, э, и, о, у, ө, ү, я, е, ё, ю, й, ы*, 20 consonants: *м, н, г, л, б, в, р, ц, ж, з, с, д, т, ш, ч, х, п, к, ф, щ*, and 2 signs that are borrowed from the Russian Cyrillic alphabet in order to express special characteristics of Mongolian pronunciation: *ь, ъ*. Moreover, there are also long and diphthong vowels based on basic seven vowels **(***а, э, и, о, у, ө, ү***)**: **(***аа, ээ, оо, уу, өө, үү, and ай, эй, ой, уй, үй, ий***)**. Thus, the number of total different characters for the alphabet is forty-seven.

Mongolian vowels are classified into masculine **(***а, о, у, я, ё, ю*(*у*)*, ы***)**, and feminine vowels **(***э, ө, ү, е, ю*(*ү*)**)**, and neutral vowels **(***и, й***)**. According to the vowel harmony each Mongolian word may have only masculine or feminine vowels. Neutral vowels can appear with either set. But, the rule of vowel harmony is not applied to word composition or to verbs with inflectional suffixes of past simple tense '*-жээ*' [*-jee*], and '*-чээ*' [*-chee*].

	үйл-д-вэр-л(э)-л
үйл	noun-stem meaning "act, deed"
-д	verb building suffix; output "үйлд" [uild] has meaning "produce"
-вэр	noun building suffix; output "үйлдвэр" [uildver] has meaning "factory"
-л(э)	verb building suffix; output "үйлдвэрлэ" [uildverle] has meaning "produce"
-л	noun building suffix; output "үйлдвэрлэл" [uildverle] has meaning "manufacture"

Fig. 2. Distinction of morphemes

A word is a basic unit for a morphological analysis performance. Mongolian words are relatively easily detected from the text since a space is supposed to be placed between them. Mongolian words are composed of at least two or more morphemes.

Each morpheme has its own meaning, function, quantity, and form which makes the distinction of the boundaries relatively clear. For example **(**Fig. 2**)**:

Mongolian words have plenty of morphophonemic rules, which create difficulties for their morphemic analysis.

Morphemes are used for word structure according to vowel harmony and to build a new word. For example, *яв+лаа* [*yav+laa*] **(**went - past simple tense**)**, *ир+лээ* [*ir+lee*] **(**came - past simple tense**)**, *оч+лоо* [*och+loo*] **(**went - past simple tense**)**, *өг+лөө* [*oeg+loeo*] **(**gave - past simple tense**)**. Here, the different suffixes such as '***-лаа***' [*laa*], '***-лээ***' [*lee*], '***-лоо***' [*loo*], and '***-лөө***' [*loeo*] are a result of some vowel harmony with the following four vowels: **(***а, э, о*, and *ө***)**.

Mongolian morphemes are divided into word stem, derivational and inflectional suffixes. A word stem keeps the original meaning of the word and is a basis unit of the word family. A root with one or more derivational suffixes can also be a stem. Word stems are usually at the beginning, and never appear at the end of the word. The word stem doesn't change, but it can always be declined.

Suffixal morphemes have lexical meaning and build a new word. The word building method is very important and very productive in Mongolian.

Grammatical changes are made by adding suffixes to the word stem. Each suffix expresses only a grammatical meaning. For example: '***-в***' [*-v*] inflectional suffix of the word "*унши**в***" [*unshi**v***] **(**meaning "read"**)** expresses only the past simple tense.

Contrary to some other agglutinative languages such as Turkish there are no person or number suffixes in Mongolian verbs. Suffixes related to voice, aspect, or mood can be added to verbs in the prescribed order, although it is not necessary for a suffix from each group to be present. There are no irregular verbs in the Mongolian language.

In Mongolian, there is no gender question of nouns. Noun stems can be marked for plurality, case, possessiveness etc. in the prescribed order. For example **(**Fig. 3**)**:

	ном-ууд-аас-аа
ном	noun-stem meaning "book"
-ууд	plural inflectional suffix; output "номууд"[nomuud] has meaning "books"
-аас	ablative case suffix; output "номуудаас" [nomuudaas] has meaning "from books"
-аа	possessive suffix; output "номуудаасаа" [nomuudaasaa] has meaning "from your books"

Fig. 3. Inflectional suffixes

Distinction of adjectives and nouns is a dilemma in Mongolian. Most adjectives can be used as nouns, and

nouns can perform the function of adjectives as noun modifiers in noun-noun groups or noun phrases. Therefore we can assume that adjectives have the same morphotactics as nouns, but with very little differences.

Suffixes are used to build a word or another new stem and they differ from each other by their combining to the stem.

The root is used to build a new word or inflect the word. Word building suffixes conjugate to both root and stem, and build a new word. This makes them different from inflectional suffixes. Over three inflectional suffixes are conjugated rarely to some words. Most Mongolian words have to contain less than 7 suffixes and no more than 6 suffixes ever appear in the word structure.

Noun building suffixes and verb building suffixes interlace such as ***noun*+verb+*noun*+…** and this is a unique regularity. But the last variation of word building suffixes is usually noun-building suffixes. For example, as shown in Fig. 2 and Fig. 4, four suffixes (‘***-д***’ [*-d*], ‘***-вэр***’[*-ver*], ‘***-л(э)***’ [*-le*], and ‘***-л***’ [*-l*]) are conjugated in the word “*үйлдвэрлэл*” [*uildverlel*] (meaning “manufacture”).

After derivational morphemes that may change the part-of-speech for a word, the stem takes inflectional suffixes that are applied to its final part-of-speech category. For instance, a nominal root can become a verb via a derivational suffix and then takes verbal inflections.

Noun	Verb	Noun	Verb	Noun
үйл	үйл+д	үйлд+вэр	үйлдвэр+л(э)	үйлдвэрлэ+л
act,deed	produce	factory	produce	manufacture

Fig. 4. Interlace of word building

Fig. 5 shows the word building method in Mongolian.

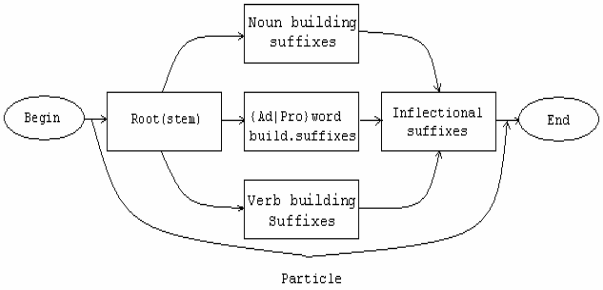

Fig. 5. Mongolian word building method

Romanization of Cyrillic Mongolian script characters is necessary for making relevant files for the analysis tool PC-KIMMO and Romanization is useful for increasing readability of the implemented rules and lexicon. Some Mongolian consonants and vowels are not expressible via a single Roman character. For example, there are at least basic seven vowel characters (*a*, *э*, *u*, *o*, *y*, *ө*, and *ү*) in Mongolian, but only five (a, e, i, o, and u) are used in the English alphabet. It means a few vowels need to be expressed by a complex form with two Roman characters. In such case, a problem would arise with regard to syllabification.

A *C++* program is created to automatically convert the 8 bit 1 or 2-byte Cyrillic Mongolian codes into relevant English alphabet characters using the table shown in Fig. 6.

Mongolian characters	а	э	и	о	у	ө	ү	я	е	ё	ю	й	ы	м	н	л	ь	г	б	в	р	ц	ж	з	с	т	ш	п	к	ф	щ	х	д	ъ	ч
Latin characters	a	e	i	o	y	oe	u	ya	ye	yo	yu	i	&i	m	n	l	^	g	b	v	r	c	j	z	s	t	sh	p	k	f		h	d	^^ch	

Fig. 6. Conversion scheme of Mongolian and English alphabet

4. Implementation of The Mongolian Rules

Two factors have to be considered in order to build a two-level morphological description using PC-KIMMO: the first structure of the lexicon and the second, the two-level rules. The lexicon represents the morphotactic description of the language. It consists of morphemes set along with labels indicating their conjugated order. The two-level rules are expressed as FSTs by state transition tables indicating symbol changes between the lexical and the surface representations.

4.1 Lexicon and Morphotactics

This sub-section shows a general description of the Mongolian lexicon and morphonemic rules. The lexicon consists of 2 kinds of files. The first one contains the list of alternations and the second one the primary lexicon. The Mongolian alternation is declared as (1):

(1) List of alternation:

```
ALTERNATION Root        N V AJ
ALTERNATION Suffix      SUFFIX
ALTERNATION Infl        INFL
ALTERNATION Noun        N SUFFIX
ALTERNATION End         End
```

The “Root” alternation states that each word in the lexicon will be either a noun (morpheme from sub-lexicon “noun”), adjective (morpheme from sub-lexicon “adjective”), or verb (morpheme from sub-lexicon “verb”). The “End” alternation blocks further looping of the suffixation and finalizes the word building process.

Each lexical entry in the lexicon is composed of at least four obligatory fields such as “lexicon form”, “sub-lexicon”, “alternation name”, and “gloss string”. The shape of lexical entries is given in (2).

The lexical form (\lf) contains lexical item (morpheme). Null item ‘0’ denotes an empty string and means the affix is optional. So that roots can be realized as words without any affixes.

The lexicon field (\lx) denotes the sub-category and the item is included in it. A pre-declared part-of-speech is specified in this field.

The gloss field (\gl) is an optional field for any additional grammatical information relevant to the morpheme. This field will show up only during the recognition process. This field is not used in this work.

The alternation field (\alt) is used to detain the

morphemes sequence. That field of **(**2a**)** means N should be followed only by a lexical item and its sub-lexicon name is one of the elements of the "Noun" group which should be specified in advance as in **(**1**)** and this is the way Mongolian noun compounds are produced.

(2) Description of lexical entries*:*

```
a:    \lf harilcaa
      \lx N
      \alt Noun
      \gl1
      \gl2
b:    \lf har^
      \lx V
      \alt Suffix
      \gl1
      \gl2
c:    \lf hyrdan
      \lx AJ
      \alt Suffix
      \gl1
      \gl2
d:    \lf 0
      \lx SUFFIX
      \alt Infl
      \gl1
      \gl2
e:    \lf 0
      \lx End
      \alt #
      \gl1
      \gl2
```

According to the above-mentioned declaration Mongolian initial morph tactics of complex words can be described as in Fig. 5.

4.2 Rules Component

Mongolian morphophonemic rules are described in this sub section. All the Romanized alphabetical characters and other symbols and diacritics such as boundaries used in the rule description should be declared at the beginning of the rule file. Then two-level rules of Mongolian morphology are described in it. Our Romanized alphabet consists of 19 consonants, 3 signs, 8 vowels, and 2 concatenators **(**+, **-)**. They are specified in **(**3**)** as along with other symbols:

(3) Description of the Romanized alphabet:

```
ALPHABET
 e y a o i u oe b g j s f n d h p r q ya
 yo ye yu &i z ch sh v c m l t + ` ^ - ^^
NULL      0
ANY       @
BOUNDARY  #
```

Null symbol is necessary for the rules of insertion or deletion of characters or boundaries, and symbol '@' is used for specifying default correspondences and elsewhere effects. The last symbol is word boundary '#'. Hard-sign **(***ъ***)**, soft-sign **(***ь***)**, and masculine vowel **(***ы***)** are denoted by **(**^^, ^, and &i**)** respectively. Dash **(-)** is used to denote a consonant which requires a vowel to follow it. For example **(**4**)**:

(4)Vowel insertion:

```
LR: od-n-       oed-n-
SR: odOno       oedOnoe
```

One more procedure is necessary for the effective description rules for Mongolian. As mentioned in Section 3, PC-KIMMO is not enough to allow natural classification of all possible characters by way of their own notations. The additional device is "subset" notation, which is designed to contain multiple sounds in a single expression. The subsets are used for the current rule description as follows (5):

(5) Subset declaration (*according to Fig. 5)****:***

```
SUBSET MG m n g l b v r
SUBSET C b g s n d f h j r z ch sh v p c m l t
SUBSET IC d j z p c s t h ch sh
SUBSET V e y a oe o i u
SUBSET Csib sh z ch
SUBSET Diff n g
SUBSET Sign ^ ^^
SUBSET FV e u
SUBSET MV y a o
```

Subset **MG** is defined to include all consonants with vowels, **C** all consonants. Also the consonants need to separate into two subsets: **IC-** containing the incomplete consonants, **Csib -** containing sibilant consonants. Moreover, also the vowels need to group into four separate subsets: **V –** containing vowels **(**e, y, a, o, i, and u**)**, **Diff** – containing different vowels **(**n and g**)**, **Sign –** containing two signs **(**^ and ^^**)**. Subset **FV** includes feminine vowels: **(**e and u**)**, **MV**- masculine vowels: **(**y, a, and o**)**. There are nine subset definitions in total.

The next step is actual rules and transition tables to write. The rule file contains two-level rules in state transition table format. The rules are created to successfully analyze all the corpus internal complex words made up of two or more morphemes.

Most of the rules about noun and verb affixation were written about below. Lexical phonology and morphology of the two part-of-speech categories and their interactions create lots of irregularities and problems in the Mongolian word building structure.

The first rule defines the default correspondences between lexical and surface symbols. It specifies the way of lexical forms that was realized as surface forms whenever they are not a part of the rule description known as feasible pairs. A default correspondence of morpheme boundary is <+:0> so it never appeared at the surface level and the others are the same in these two forms.

Currently 36 rules were written for analysis and one of them **(**Rule 2 in (14)**)** was explained in detail for illustration.

In Mongolian, there exit surface alternation of the word ended by soft-sign **(***ь***)** denoted as **(**^**)** in Fig. 6. Its surface form appears as i when the suffix begins with a vowel or consonant such as **(***a, o, y*, and *y,* or *м, н, г, л, б, в, р*, and *х***)**, while **(**^**)** appears in other environment. For example **(6)**:

(6) ь/u alternation*:*

```
;LR: doh^+oo   har^+d
;SR: dohio     har^d
;LR: har^+h    har^+lcaa
;SR: harih     harilcaa
```

The rule for this process is written as **(**7**)**:

(7) ^:i transformation rule:

```
^:i => ___+:0 [V:0|MG|h]
```

The operator **(**=>**)** indicates the condition "only but not always". Thus, the rule states that sign **(**^**)** is realized as **(**i**)** only but not always in the given environment. The subset **MG** means either of the consonants with vowels as pre-specified in **(**5**)**. The bracket symbols means any of the items within them divided by a vertical bar **(**|**)** can be placed at that position.

The rule was compiled by compiler tool "Kgen" into formation of state transition table as shown in **(**8**)**:

(8) Version 1:

```
RULE "^:i => ___+:0 [V:0|MG|h]" 3 6
      ^  +  V  MG  h  @
      i  0  0  MG  h  @
 1:   2  1  1   1  1  1
 2.   0  3  0   0  0  0
 3.   0  0  1   1  1  0
```

Only the environment of <^:i> was defined in **(**8**)**. The environment of the transformation <^:^> is not described in the rule. Therefore, such state is added into the rule **(**8**)**, so the rule is modified as **(**9**)**:

(9) Version 2:

```
RULE "^:i => ___+:0 [V:0|MG|h]" 4 7
      ^  ^  +  V   MG  h  @
      i  ^  0  0   MG  h  @
 1:   2  3  1  1   1   1  1
 2.   0  0  3  0   0   0  0
 3.   0  0  4  1   1   1  0
 4.   0  0  0  0   0   0  1
```

In Mongolian, a short vowel for a short vowel ended word **(**for example: "*darg**a***"**)** or consonant following of short vowel ended word **(**for example: "*zur**a**g*"**)** has to be dropped **(**for example: *darga+aa=darg0aa*, *zurag+iin= zur0giin***)** when a suffix beginning with long vowels (for example: '**-aa**' and '**-iin**') is added to it. That case is expressed by appending new states into **(**10**)** instead of <V:0>. Thus, the rule was modified to **(**11**)**:

(11) Version 3:

```
RULE
"2 ^:i-spelling and vowel dropping" 6 14
   ^ ^ + a o e y i u &i oe h MG @
   i ^ 0 0 0 0 0 0 0 &i 0  h MG @
1: 2 3 0 5 5 5 5 5 5 1  5  1 1  1
2. 0 0 3 0 0 0 0 0 0 0  0  0 0  1
3. 0 0 4 1 1 1 1 1 1 1  1  1 1  0
4. 0 0 0 1 1 1 1 1 1 1  1  0 0  1
5. 0 0 1 0 0 0 0 0 0 0  0  6 6  6
6. 0 0 1 0 0 0 0 0 0 0  0  0 0  0
```

There would appear lots of irregularities or problems besides considered states in the given environment of **(**11**)**. For instance, the sequence of 3 vowels or soft sign ending vowel or vowel ending soft sign is not allowed in a word. Moreover some mistakes would appear in it as the vowel is not dropped when soft sign **(**^**)** is changed to **(**i**)** and vice versa. These exceptional cases can be resolved by adding other rules, which prohibit occurrence of such irregular forms **(**Rules 3-7 in (14)**)**. Some rules are shown in **(**12**)**.

A revision to prohibit the above irregularities was made by hand in **(**12**)**:

(12) Prohibition rules:

```
;Prohibition rule for ^:i spelling
;LR:  har^+aad
;SR:  har^+0ad
RULE "3 +:0 /<= ^___V:0" 3 4
      ^  +  V   @
      ^  0  0   @
 1:   2  1  1   1
 2:   2  3  1   1
 3:   2  1  0   1

;Prohibition rule for vowel dropping
;LR: olon+oos olon+oos
;SR: ol0n0oos olon0oos

RULE "7 +:0 /<=V:0 C___C" 4 4
      V  C  +   @
      0  C  0   @
 1:   2  1  1   1
 2:   2  3  1   1
 3:   2  1  4   1
 4:   2  0  1   1
```

Outputs of this rule are as follows **(**13**)**:

(13) Test results:

```
;===========
;^:i spelling
;===========
;LR: har^+aad  har^+lcaa orh^+ood  bar^+h har^+d
;SR: hari00ad  hari0lcaa orhi00od  bari0h har^0d
;LR: doh^+oo   doh^
;SR: dohi00o   dohio

;===========
;Vowel dropping
;===========
;LR: olon+oos hyral+yyd  gazar+aar
;SR: ol0n0oos hyr0l0yyd  gaz0r0aar
;LR:garga+aad orgo+ood olon+&ig darga+&ig
;SR:garg00aad org00ood ol0n0&ig darg00&ig
```

The file Mongolian.rul contains thirty-six rules in the state transition table format of two-level rules:

(1**)** Rule 1 defines the default corres-pondences between lexical and surface symbols. Here, the default surface realizations are the same as their lexical representations, whereas the symbol **(**+**)** is mapped to zero. Therefore, the lexical form of a word "*яв+сан*" [*yav+san*] **(**go+ past simple tense**)** will map to the surface form "*яв*0*сан*" what is displayed as "*явсан*" [*yavsan*] **(**went**)**, because zero is not printed in the surface form.

(2**)** Rule 2 defines the state that a trans-formation of soft-sign **(***ь***)** into **(***и***)** would occur when the suffix begins with a vowel or consonant such as **(***а, о, у,* and *ү, ө***)**, or **(***м, н, г, л, б, в, р,* and x**)**. Moreover, this rule can be applied to vowel dropping. For example: *orh^+ood=orhi0od*, *garga+aad=garg0aad*, *olon+oos=ol0noos*

(3-7**)** Rules 3-7 prohibit irregular cases, which were motivated by Rule 2. There are irregular forms, such as a vowel is not dropped when a soft sign changed into **(**i**)** and vice versa.

(8**)** Rule 8 states that a vowel insertion would occur when adding a suffix '***-н***' to each of **(***ж, ч*, and *ш***)** ended words. For example: o*ch+n=ochin*

(9**)** Rule 9 states that a soft-sign **(**ь**)** insertion would appear when adding '**-е**' suffix to a feminine word. For example: *ug+ye=ug^ye*

(10-15**)** Rules 10-15 state that vowel insertion would occur according to vowel harmony when adding an incomplete consonant-began suffixes to words ended by an incomplete consonant. For example: *dyrs+j=dyrsaj*

Some states of these rules also are not allowed the occurrence of irregular forms produced in association with this process.

(16-20**)** Rules 16-20 describe the vowel insertion process. These rules declare that a vowel has to be inserted when a suffix '***-н***' is added to a word ended with two conjugated consonants and they prohibit the irregularities during the process. For example: *hyrald+n=hyraldan*

(21-22**)** Rules 21-22 state that hard sign **(**ъ**)** insertion would occur when adding a suffix '***-я***' or '***-ё***' to a consonant ended masculine word. For example: *orold+yo=orold^^yo, argad+ya=argad^^ya*

(23-36**)** Rules 23-36 state that a vowel would be inserted when an alveolar ridge **'*-н***' suffix is conjugated to a word and this vowel insertion was marked by dash **(-)**. Also some irregularities are prohibited in these rules. For example: *todor-n- = todor0no*

The above rules are listed in the **(**14**)**. All rules interact with each other and they are identical. That may result in some rules being difficult to modify without a careful investigation of other rules in the list.

(14) List of Rules:

```
RULE "Defaults 1" 1 37
     b g s n d h f r v c m l p j q ^^
     b g s n d h f r v c m l p j q ^^
1:   1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

     ya yo ye yu ch sh t e y a o i u
     ya yo ye yu ch sh t e y a o i u
     1  1  1  1  1  1  1 1 1 1 1 1 1
     oe z + ` ^ &i - @
     oe z 0 0 ^ &i 0 @
     1  1 1 1 1 1  1 1
;===========
;^:i spelling and vowel dropping
;===========
;LR: har^+aad  har^+lcaa bar^+h
;SR: hari00ad  hari0lcaa bari0h
;LR: olon+oos  olon+&ig orgo+ood
;SR: ol0n0oos  ol0n0&ig org00ood

RULE
"2 ^:i-spelling and vowel dropping" 6
14

    ^ ^ + a o e y i u &i oe h MG @
    i ^ 0 0 0 0 0 0 0 &i 0  h MG @
 1: 2 3 0 5 5 5 5 5 5 1  5  1 1  1
 2. 0 0 3 0 0 0 0 0 0 0  0  0 0  1
 3. 0 0 4 1 1 1 1 1 1 1  1  1 1  0
 4. 0 0 0 1 1 1 1 1 1 1  1  0 0  1
 5. 0 0 1 0 0 0 0 0 0 0  0  6 6  6
 6. 0 0 1 0 0 0 0 0 0 0  0  0 0  0
;LR: har^+aad har^+aad har^+aad
;SR: hari00ad har^0aad hari00ad
RULE "3 +:0 /<= ^___V:0"
RULE "4  +:0 /<=  ^___V V"
RULE "5 +:V /<= _V V"
;vowel dropping rule
;LR: olon+oos olon+oos
;SR: olon0oos ol0n0oos
RULE "7 +:0 /<=V:0C__C"
;====================
;vowel insertion rule
;====================
;LR: ynsh+n, ysch+n
;SR: ynshin, yschin
RULE "8 " 7 6
     +   Csib  +    n    #    @
     i   Csib  @    n    #    @
 1:  0    2    1    2    1    1
 2:  3    2    5    2    1    1
 3.  0    0    0    4    0    0
 4.  0    0    1    0    1    0
 5:  0    1    1    6    1    1
 6:  0    1    0    1    0    1
 7:  3    2    1    2    1    1
;LR: end+ye uns+ye
;SR: end^ye uns^ye
RULE "9 +:^ =>FVC(C)__ye"
;LR: bos+j hot+sh noc+j orold+c
;SR: bosoj hotosh nocoj oroldoc
RULE "10 +:o <=> o(C)IC__IC"
;LR: oergoegd+j
;SR: oergoegdoej
RULE "11 +:oe <=>oe(C)IC__IC"
;LR: hevt+j uz+j
;SR: hevtej uzej
RULE "12 +:e <=>FV(C)IC__IC"
;LR: byh+j ynt+j yn+j
;SR: byhaj yntaj ynaj
RULE "13 +:a <=>[a|y](C)IC__IC"
;LR: yn+j yn+j
;SR: yn0j ynaj
RULE "14 +:0 /<= IC__IC"
RULE "15 +:0 /<=IC__l"
;LR: orold+n
;SR: oroldon
```

```
RULE "16 +:o <=o C C__n"
;LR: hyrald+n hyrd+n
;SR: hyraldan hyrdan
RULE "17 +:a <=[a|y]C C__n"
;LR: oergoegd+n
;SR: oergoegdoen
RULE "18 +:oe <= oe C C__n"
;LR: hevt+n
;SR: hevten
RULE "19 +:e <=[e|u]C C__n" 5 7
RULE "20 +:0 /<=C C__n"
;LR:gar+ya  or+yo
;SR:gar^^ya or^^yo
RULE "21 +:^^<=>[a|o]C(C)__[ya|yo]"
RULE "22 +:0 /<= C__[ya|yo]"
;LR: od-n-
;SR: od0no
RULE "23 -:0 <=o(C)C__n -:o"
;LR: oed-n-
;SR: oed0noe
RULE "24 -:0 <=oe(C)C__n -:oe"
;LR: yd-n- had-n-
;SR: yd0na had0na
RULE "25 -:0 <=[a|y](C)C__n -:a"
;LR: uz-n-  er-n-
;SR: uz0ne  er0ne
RULE "26 -:0 <=[e|u](C)C__n -:e"
RULE "27 -:0 /<=V(C)C__n -:0"
RULE "28 -:V /<= V (C) C__n"
RULE "29 -:0 /<=FV(C)C__n -:a"
RULE "30 -:0 /<=MV(C)C__n -:FV"
RULE "31 -:0 /<=o(C)C__n -:a"
RULE "32 -:0 /<=a(C)C__n -:o"
RULE "33 -:0 /<=e(C)C__n -:oe"
RULE "34 -:0 /<=[a|y](C)C__n -:oe"
RULE "35 -:0 /<=o(C)C__n -:oe"
RULE "36 -:0/<=oe(C)C__n[-:FV|-:MV]"
```

5. Results And Discussion

The implementation of a two-level morphological description for the Mongolian language was based on PC-KIMMO software. A modern Mongolian novel and a translated novel from a foreign language were used for checking the performance of rules that were constructed by us. A list of unrepeated words was generated from above texts by changing all the spaces into new lines followed by conversion to Romanization.

After the PC-KIMMO is opened, the rule files, sub-lexicons, and grammar descriptions are loaded file by file, then a test file or individual words are tested on Pentium IV with 2.66 GHZ speed and 256 MB RAM. The sub-lexicons consist of 6,199 nouns, 18,551 verbs, 4,516 adjectives, and 223 affixes.

By testing the two-level morphological description, a total of 63 % of 4,105 unrepeated words were recognized for the Mongolian novel in 2 seconds and 58 % of 7,595 unrepeated words from the translated text in 4 seconds. For the conclusion, the recognition results were required more development on rule processing beside of the lexicon is insufficient by comparing to active Mongolian words of over 70,000.

Also, old and Cyrillic Mongolian grammars disagree in occasional cases, so that had an influence on the recognition percentage. For example, when adding a long vowel beginning affix to some nominated nouns, its last vowel is never dropped in Cyrillic Mongolian. So it causes some problems for the above mentioned rule (3-7) because PC-KIMMO couldn't recognize word classification. Although many loan words can make rule prohibitions. Therefore, recognition results depended on which kind of noun and how many loan words were used for issuing text.

Finally, it was the first step towards development of a full description for Mongolian morphology implemented in the two-level rules. Thus, it would be constantly updated and modified for further usage on application of the syntactic parsing systems.

References

[1] Alam, Y.S., 'A Two-level Morphological Analysis of Japanese' Texas Linguistic Forum, Vol. 22, pp. 229-252, 1983.

[2] Antworth, Evan, 'PC-KIMMO: A Two-level Processor for Morphological Analysis', Summer Institute of Linguistics, Inc., 1990.

[3] Damdinsuren, Ts., Osor, B. 'Dictionary of Mongolian Grammar' Ulaanbaatar, 1983.

[4] Gazdar, G., 'Review Article: Finite State Morphology' Linguistics Journal, Vol.23, pp. 597-607, 1985.

[5] Karttunen, Lauri and Kenneth Beesley, 'Two-level rule compiler', Technical Report ISTL-92-2, Xerox, Palo Alto Research Center, California, 1992.

[6] Karttunen, Lauri, Kimmo Koskenniemi, and Ronald Kaplan, 'A compiler for two-level phonological rules. In Tools for Morphological Analysis', Center for the Study of Language and Information, Stanford University, Cali-fornia, pp. 1-61, 1987.

[7] Koskenniemi, Kimmo, 'Two-level Morphology: A General Computational Model for Word-form Recognition and Production", Publications, Vol.11, University of Helsinki, Helsinki, Department of General Linguistics, 1983.

[8] Oflazer, K., 'Two-Level Description of Turkish Morphology', Literary and Linguistic Computing, Vol. 9, pp. 137-148, 1994.

[9] Purev J., Hyun Seok Park, Altangerel Ch. 'Tree adjoining grammars for Mongolian' Proceedings of 3-rd International Conference on East-Asian Language Processing and Internet Information Technology, EALPIIT2003, Ulaan-baatar, Mongolia, pp. 321-323, 2003.

[10] Ritchie, G., 'The Generative Power of Two-Level Morphological Rules', EACL-4, pp. 51-57, 1989.

[11] Russel, G.J., Pulman, S. G., Ritchie, G. D., and Black, A. W., 'A Dictionary and Morphological Analyser for English' COLING '86, pp. 277-279, 1986.

Purev Jaimaa
BS & MS, Academy of Economy, Poland (1979)
Ph.D., Mongolian University of Science and Technology (1994)
Visiting Professor, Sejong University, Korea (2002-2003)
Professor, School of Information Technology, National University of Mongolia (present)
Research Field is Natural Language Processing

Tsolmon Zundui
BS, School of Information Technology, National University of Mongolia (2004)
MS Candidate, Seoul National University (present)
Research Field is Natural Language Processing.

Altangerel Chagnaa
BS, Dept. of Electronics, National University of Mongolia (2001)
MS, School of Information Technology, National University of Mongolia (2003)
Ph.D. candidate, School of Comp. Eng. and IT, University of Ulsan (present)
Research Field is Natural Language Processing and Machine Learning.

Cheol-Young Ock
BA, Dept. of Computer Engineering, Seoul National University (1982)
MA, Dept. of Computer Engineering, Seoul National University (1984)
Ph.D., Dept. of Computer Engineering, Seoul National University (1993)
Visiting Professor, RUSSIA TOMSK Institute (1994), GLASGOW University (1996)
Professor, School of Comp. Eng. and IT, University of Ulsan (present)
Research Field is Natural Language Processing, Machine Learning, Knowledge Engineering, Ontology.