Two-Level Description of Kazakh Morphology

Harun Reşit Zafer
Department of Computer Eng.
Fatih University
hrzafer@fatih.edu.tr

Birol Tilki
Department of Computer Programming
Vocational School, Fatih University
birolt@fatih.edu.tr

Atakan Kurt
Department of Computer Eng.
Fatih University
akurt@fatih.edu.tr

Mehmet Kara
Deparment of Contemporary Turkic Lang.
Istanbul University
mehkara@yahoo.com

Abstract: Koskemnieni's two-level model has received a lot attention in modeling morphology. In this paper we present an ongoing study on a comprehensive two-level description of Kazakh morphology. Our description is implemented using the morphological parser in the Dilmaç Machine Translation Framework. A lexicon containing the root words of contemporary Kazakh is used in the testing. Phonological and morphological special cases and exceptions have been considered in nominal, and verbal conjugations. To out knowledge this is the first time Kazakh phonological rules and morphotactics are computationally described which makes it possible to implement other linguistics applications such as machine translation systems.

Keywords: Kazakh, two-level morphology, orthographic rules, finite state machines.

Introduction

Two-level morphology [[182]] has been applied to many languages. Tools to implement two-level morphology such as PC-KIMMO [[183]] is publicly available. It was originally applied to describe finite state Finnish morphology by Koskenniemi. A detailed description with an application to English is given by Antwort [[184]]. Two-level or finite state models later were applied to many languages such as Japanese [[185]], Korean [[186]], Turkish [[187]], Arabic [[188]], and Mongolian [[189]]. All these languages except Arabic are related linguistically. They are Altaic languages. Like Ural languages of Finnish and Hungarian they are agglutinative.

There is a group of languages called Turkic Languages including Turkish, Turkmen, Kazakh, Uzbek, Kyrgyz, Azerbaijani. There are more than 20 languages in this group. These languages share a lot in common from phonological, morphological and syntactic aspects. However they are not intelligible for the most part.

[182] Koskenniemi, K., 1983, Two-Level Morphology: A General Computational Model of word-form recognition and production, Tech. Rep. Publication No. 11, Department of General Linguistics, University of Helsinky.

[183] Karttunen L, 1983, PC-KIMMO: A General Morphological Processor. In Texas Linguistics Forum 22, pp.165-186.

[184] Antworth, E.L., 1990, PC-KIMMO: A Two-level Processor of Morphological Analysis, Summer Instıtute of Linguistics, Dallas, TX.

[185] Alam, Y.S., 1983, Two-level Morphological Analysis of Japanese, Texas Linguistics Forum 22, pp. 229-252.

[186] Kim, D. B., Lee S. J., Choi, K.S., and Kim, G.C., 1994. A two-level morphological analysis of Korean. In **Proceedings of the 15th conference on Computational linguistics - Volume 1** (COLING '94), pp. 535-539.

[187] Oflazer, K. 1994, Two-level description of Turkish morphology, Literary and Linguistic Computing, Literary and Linguistic Computing Volume9, Issue2 pp. 137-148.

[188] Arabic Finite State Morphological Analysis and Generation, In COLING-96, Cophenagen, pp. 89-94.

[189] Jaimai, P., Zundui, T., Chagnaa, A., and Ock, C.Y., PC-KIMMO-based Description of Mongolian Morphology, International Journal of Information Processing Systems Vol.1, No.1, 2005 pp. 41-48.

They are mostly spoken in Turkey, Turkic states, in Central Asia and in various parts of Russia and other parts of the world.

These languages are except Turkish are usually resource poor from computational linguistics point of view. Although recently there are some work on the Turkmen [[190] [191]], Azerbaijani, Uyghur and others. It can be said that we are only at the beginning of research considering the many languages in this group. Kazakh is one of the important languages in this group considering the number of people speaking this language.

Kazakh (also Qazaq) language is a Turkic language and belongs to Kypchak branch. It is the official language of Kazakhstan. It is spoken about 12 million people all over the world. Like other Turkic Languages Kazakh is also agglutinative and employs vowel harmony [[192]].

This paper is organized as follows: In Section 2 Kazakh orthography is described using two-level rules of Koskenniemi. Kazakh alphabet and phonological rules are defined here. In Section 3 we briefly discuss Kazakh morphotactics using Finite State Machines with a few examples. Conclusions and future work is given in the last section.

Kazakh Orthography

Kazakh is officially written in the Cyrillic alphabet. We will use a latin transcription of Cyrill version for convenience. There is a transliteration system converting from Kazakh Cyrill to Latin [[193]]. Kazakh alphabet is given in Table 1 shows current Kazakh Alphabet and its transliteration to Latin Alphabet.

Table 1: Cyrillic Kazakh alphabet and its transliteration to Latin alphabet.

Cyrillic	Latin	Cyrillic	Latin	Cyrillic	Latin
Аа	Aa	Ққ	Qq	Фф	Ff
Әə	Ää	Лл	Ll	һ	H
Бб	Bb	Мм	Mm	Хх	Xx
Вв	Vv	Нн	Nn	Цц	Tsts
Гг	Gg	ң	Ñ	Чч	Çç
Ғғ	Ğğ	Оо	Oo	Шш	Şş
Дд	Dd	Өө	Öö	Щщ	Şçşç
Ее	Eye	Пп	Pp	Ыы	Iı
Ёё	Yoyo	Рр	Rr	Іі	İi
Жж	Jj	Сс	Ss	Ээ	Ee
Зз	Zz	Тт	Tt	Юю	Yuyu
Ии	İyiy	Уу	Uwuw	Яя	Yaya
Й	Y	Ұұ	Uw		
Кк	Kk	Үү	Üü		

Two-level morphology is a language-independent method to model morphologic rules of natural languages. In this model words are represented in two forms; lexical and surface. Two-level rules define transformation between the two forms. Phonological rules in this model can be expressed in one the following formulations:

a:b => LC__RC

This rule states that a lexical a, corresponds to a surface b only if it follows the left context (LC) and/or precedes the right context (RC). This correspondence only occurs under this condition but not always.

a:b <= LC__RC

This rule states that a lexical a, always corresponds to a surface b if it follows the left context (LC) and/or precedes the right context (RC). This correspondence always occurs with this condition but can also occur with different conditions.

[190] M. Shylov, “Dilmaç: Turkish and Turkmen Morphological Analyzer and Machine Translation Program,” Master’s thesis, Fatih University, İstanbul Turkey, 2008.

[191] Tantuğ, A. Cüneyd and Adalı, Eşref and Oflazer, Kemal (2006) **Computer analysis of the Turkmen language morphology.** Advances in natural language processing, proceedings (Lecture notes in artificial intelligence), 4139 . pp. 186-193.

[192] Dzhubanov, A., Khasanov, B.. 1973. Computational description of the Kazakh language. In **Proceedings of the 5th conference on Computational linguistics - Volume 2**(COLING '73), Vol. 2. Stroudsburg, PA, USA, 75-77.

[193] Buran, A., Alkaya, E. (in Turkish) “Çağdaş Türk lehçeleri,” ANKARA: Akçağ, 2009, pp. 273-312.

a:b <=> LC__RC

This states that a lexical a, corresponds to a surface b always and only if it follows the left context (LC) and/or precedes the right context (RC). This correspondence never occurs with any other condition.

a:b \<= LC__RC

This rule states that a lexical a, never corresponds to a surface b in given the environment of left context (LC) and right context (RC). This correspondence never occurs under this condition.

Below we present a set of meta-phonems used in expressing rules. The Latin Kazakh alphabet consists of 30 letters. There are 9 vowels and 21 consonants in this alphabet. The letter groups that used in rules are defined below:

Consonants: C= {b, g, ğ, d, j, z, y, k, q, l, m, n, ñ, p, r, s, t, w, x, h, ş }
Vowels: V= {a, ä, e, ı, i, o, ö, u, ü}
Back Vowels: V_b = {a, ı, o, u }
Front Vowels: V_f = {e, ä, i, ö, ü}
I = {ı, i}
A = {a, e}
L = {l, d, t}
Q = {ğ, q}
G = {k, g}
K = {k, q}
M = {m, b, p}
N = {n, d, t}
D = {d, t}
S = {s}.

There are two different lexical s in Kazakh. The S is used for the one that is never deleted on the surface form. And letter s is used for the one that can be deleted on the surface form under some conditions.

2.2 Two Level Orthographic Rules

Kazakh has the most strong vowel harmony among Turkic languages [194]. Vowels in a suffix have to agree with the preceding morpheme's vowels. Consonant harmony or assimilation is also strong in Kazakh. [195] Voiced consonants are converted into voiceless ones or vice versa. Consonants can be assimilated by preceding consonants or vowels. Under certain circumstances sound dissimilation can occur. When concatenating a morpheme to a stem, consonants or vowels can be deleted. The deleted letters can belong to either stem or suffix.

Below are some of the two-level morphologic rules of Kazakh language. We consulted the following language resources on morphology [194, 195, 193] in creating these rules. We give only a portion of the rules because of space limitation.

1. k:g <=> V __ +:0 (@:0)V
2. q:ğ <=> V __ +:0 (@:0)V
3. p:b <=> V __ +:0 (@:0)V

The consonants k, q and p at the end of stem are converted to g, ğ and b respectively when the preceding letter and the first letter of affixed morpheme are vowels.

Lexical: jürek+sI N(heart)+Poss3PS
Surface: jüreg0i jüregi (his heart)

Lexical: ayaq+sI+nDA N(foot)+Poss3PS+Loc
Surface: ayağ0ında ayağında (on his leg)

[194] Tamir, F., (in Turkish) "Kazak Türkçesi," Türk Lehçeleri Grameri, ANKARA: 2007, pp. 430-480.
[195] Koç, K., Doğan, O., (in Turkish) Kazak Türkçesi Grameri, ANKARA: Gazi Kitabevi, 2004.

Lexical: kitap+sI	N(book)+Poss3PS
Surface: kitab0ı	kitabı (his book)

4. L:d <=> [l | m | n | ñ | z | j] +:0__

The lexical L at the beginning of affixed morpheme is converted to d when the last letter of stem is one of consonants l, m, n, ñ, z and j.

Lexical: jol+LAr	N(road)+PLU
Surface: jol0dar	joldar (roads)

Lexical: beyne+LA	N(shape)+NtoV
Surface: beyne0le	beynele V(shape)

Lexical: söz+Lik	N(word)+NtoN
Surface: söz0dik	sözdik (dictionary)

5. L:t <=> [k | q | p | s | t | ş | ç] +:0 __

The lexical L at the beginning of affixed morpheme is converted to t when the last letter of stem is one of voiceless consonants k, q, p, s, t, ş, ç. Otherwise L is converted to l by default.

Lexical: ädep+LI	N(manners)+NtoADJ
Surface: ädep0ti	ädepti (well-mannered)

Lexical: tas+LAr	N(stone)+PLU
Surface: tas0tar	tastar (stones)

Lexical: Qazaq+LAr	N(Qazaq)+PLU
Surface: Qazaq0tar	Qazaqtar (Qazaqs)

7. V:0 => V+:0__

If both last letter of the word and first letter of the suffix are vowels then the first letter of suffix is deleted.

Lexical: bala + Im	N(çocuk) + Poss1PS
Surface: bala0m	balam (my child)

Lexical: caqında + Ip	V(get closer) + VtoADJ
Surface: caqında0p	caqındap (by getting closer)

8. s:0 <=> C +:0__

An s at the beginning of the suffix is deleted when the word end with a consonant.

Lexical: jürek+sI	N(heart)+Poss3PS
Surface: jüreg0i	jüregi (his/her/its heard)

Finite State Morphotactics

In agglutinative languages morphemes are affixed to the root successively. This affixation is dependent on the morphotactic rules of the language. Morphotactic rules define the suffixes that can be added to a word in a certain state. Each suffix changes the state of word that it is affixed. Morphotactic rules can be represented by a finite state machine.

A finite state machine, which in principal is a directed graph, consists of a set of states and a set of transitions among these states. Transitions are the edges of graph labeled with inflectional or derivational morphemes defining in what order those morphemes can be affixed to a word. The immediate states, in a way, represent partial words and their part of speech tagging. The initial states represent the roots words from a lexicon and their part of speech such as noun, verb, adverb, adjective, etc. The final states represent full words created by starting with a root word in an initial state and affixing morphemes on the transitions to the partial words in each intermediate state. We define the nominal, verbal and adverbial morphotactics of the language using this FSM model.

The initial states such as noun or verb are represented by rectangles. Each state is shown by rounded rectangles. The end states are defined by double bordered circles. The states represented by dotted rounded rectangles can not be a solution for words. But all other states can be a solution.

Here is an example inflection of the noun *üy (house).*

```
house  +N         +PL    +P3Sg  +LOC   +REL  +GEN
üy     +0         +LAr   +sI    +ndA   +Gi   +NIñ
üy     +0         +ler   +i     +nde   +gi   +niñ
```
üylerindeginiñ (of the thing in their house - evlerindekinin)

In this nominal analysis the following nodes are visited in FSA: Noun, Plural, Possesive 3rd Person Single, Locative case, Relative,Genitive.

Here is a verbal inflection example in Kazakh:

```
bar    +Ma  +Qan  +Min
arrive +NEG +PAST +P1s
bar    +ma  +ğan  +mın
```
barmağanmın ((I was told) I hadn't arrive)

The following nodes in FSA are visited in this analysis: Verb, Negative, Indefinite Past, 1st Person Single.

```
bar     +AtIn +0 MA +0 edi +m
arrive  +FUTR +QUE  +PAST  +P1s
bar      +atın +0 ba +0 edi +m
```
baratın ba edim (Was I going to arrive)

The following nodes in FSA are visited in this analysis: Verb, Future tense, Question, Past Continuous, 1st person single.

```
bar    +UwIm kerek +0 emes +0 bolsa
arrive +NECS P1s   +NEG    +COND
bar    +uwım kerek +0 emes +0 bolsa
```
baruwım kerek emes bolsa (If I shouldn't arrive)

The following nodes in FSA are visited in this analysis: Verb, Necessity for 1st person, Negative, Condition.

Conclusions

A comprehensive description of Kazakh Language is given using Koskemnieni two level morphology for the first time. We described the Kazakh phonological system using 27 two level rules which describes the mapping between lexical level and surface level of a word. Then we use the finite state machines to define nominal and verbal morphotactics. We implemented both orthographic rules and the finite state morphotactics on Dilmaç Machine Translation Framework [[190]]. Dilmaç is a language independent framework. Language specifications are represented in XML files in Dilmaç. No programming is required. System is web based and our implementation can be found on the Internet.

Currently we are implementing a Kazakh-Turkish Machine Translation System on Dilmac. Since both languages in the same language family, they have a lot in common from phonological, morphological and syntactic aspect. Phonological and syntactic differences generally do not pose any significant problems and can be handled easily. However two languages have different morpheme sets and lexicons. A morphological word by word translation requires a morphological parsing in source language (Kazakh), a bilingual translation dictionary to translate word stems into target language (Turkish), and a morphological generator to generate the translation by affixing the morphemes the word stem in the proper order.

Acknowledgement

We are grateful to Dr. Kalmamat Kulamshaev for his valuable insight in this study.