Finite State Approach to the Kazakh Nominal Paradigm

Bakyt M. Kairakbay
K.I.Satpayev Kazakh National
Technical University, National Open
Research Laboratory of Information
and Space Technologies / Satpayev
str., 22, Almaty 050000, Republic of
Kazakhstan
bkairakbay@gmail.com,
b.kairakbay@norlist.kz

David L. Zaurbekov
K.I.Satpayev Kazakh National
Technical University, National Open
Research Laboratory of Information
and Space Technologies / Satpayev
str., 22, Almaty 050000, Republic of
Kazakhstan
d.zaurbekov@norlist.kz

Abstract

This work presents the finite state approach to the Kazakh nominal paradigm. The development and implementation of a finite-state transducer for the nominal paradigm of the Kazakh language belonging to agglutinative languages were undertaken. The morphophonemic constraints that are imposed by the Kazakh language synharmonism (vowels and consonants harmony) on the combinations of letters under affix joining as well as morphotactics are considered. Developed Kazakh finite state transducer realizes some morphological analysis/generation functions. A preliminary testing on the use of the morphological analyzer after OCR preprocessing for correcting errors in the Kazakh texts was made.

1 Introduction

Morphological transformations of words of a natural language are relevant to many application areas relating to information processing. Finite state methodology is sufficiently mature and well-developed for use in a number of areas of natural language processing (NLP). This paper presents the development and implementation of a finite-state transducer for a nominal paradigm of the Kazakh language. The morphophonemic constraints that are imposed by the Kazakh language synharmonism (vowels and consonants harmony) on the combinations of letters under affix joining as well as morphotactics are considered. Then on the basis of the nominal paradigm formalization it is possible to build a computer implementation of morphological analysis/synthesis of word forms (morphological module) with using of two-level morphology (Koskenniemi, 1983) and finite state morphology (Beesley and Karttunen, 2003). Our morphological module, in turn, is the part of larger Web 2.0 service-oriented system of the Kazakh text recognition involving the Kazakh OCR-module (Kairakbay et al, 2012).

The finite state and two-level morphology approach has been used successfully in a broad number of NLP applications including agglutinative languages. Among them one can be noted the Basque language (Alegria et al, 1996, 2002) belonging to ergative-absolutive languages with agglutinative features, nonconcatenative Arabic language (Beesley, 2001; Attia et al, 2011), pure agglutinative languages as Turkish (Oflazer, 1994, 1996; Eryiğit and Adalı, 2004; Çöltekin, 2010), Turkmen language (Tantuğ et al, 2006), Crimean Tatar language (Altintas and Cicekli, 2001), Uygur language (Orhun et al, 2009; Wumaier et al, 2009), Kyrgyz language (Washington et al, 2012), Kazakh language (Altenbek and Wang, 2010), and many others. Last cited paper studies the Kazakh as minority language in Xinjiang of China where obsolete alphabet based on Arabic notations is used, so that is just indirectly related to our research.

The Kazakh language (Baskakov et al, 1966; Krippes, 1996; Mussayev, 2008) belongs to Ural-Altaic family of agglutinating languages (Baskakov, 1981). In such languages the concept of the word is much broader than simply a set of items of vocabulary. As an illustration for the inflectional paradigm of the Kazakh language let's give the following example a Kazakh word (Mussayev, 2008): "*ata +lar +ymyz +da +ġy +lar+dìkì+n*"[1] that is equivalent to English sentence "*that there is at the items belonging to our fathers*".

[1] Further we follow the notations of ISO 9 (1995) for the transliteration of modern Kazakh letters.

Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, pages 108–112,
St Andrews–Sctotland, July 15–17, 2013.

Historically a variety of alphabets had been used for the Kazakh language. Arabic alphabet was used from the tenth century until 1929. This alphabet is still prevalent in Xinjiang, China and in some Kazakh Diaspora abroad. From 1929 to 1940 there was the Latin alphabet which was replaced then on the Cyrillic alphabet. Modern Kazakh alphabet based on Cyrillic contains 33 standard Cyrillic characters of Russian and 9 additional characters that reflect specific sounds of the Kazakh language.

2 The Kazakh Nominal Paradigm

Morphophonemics of the Kazakh language can be expressed by the following set of rules.

Vowels harmony

Consonance of syllables

- Vowel is a syllable-forming element;
- The number of syllables in a word is equal to the number of vowels in the word;
- Vowel determines a type of word: original Kazakh words are either only the back or front only;
- If the preceding syllable contains a back (front) vowel the appended affixes should be back (front);
- Exceptions apply to the borrowed words only.

Consonance of sounds

- Consonants *ġ, ķ, h, ḥ* are combined only with back vowels *a, o, u̇, y*;
- Consonants *g*, *k* are combined only with front vowels *a̋, ô, ù, ì, e*.

Consonants harmony

Progressive assimilation: a subsequent consonant has become like the preceding consonant on the syllable boundary;

Regressive assimilation: the subsequent sound affects to the preceding one.

The order of attachment of inflectional affixes (morphotactics) to the Kazakh stem is as follows:

Stem + plural affix + possessive personal affix + possessive abstract affix + case affix

The choice of concrete surface form of affix formal representation is determined by the phonological rules, i.e. by the vowels and consonants harmony. Then:

- Plural affix is appended directly to the stem. Singular is determined by the absence of plural affix;
- Possession affix is placed after plural affix (if any);
- Plural and possessive affixes can be swapped for collective nouns;
- Case affixes that are located after the plural and possessive affixes are the same for all categories of nouns;
- Possession affix -*Dìkì/(n)ìkì* can append additionally after other possession affixes under predicative substantivation.

Total number of formulated rules for the Kazakh nominal paradigm morphophonemics and morphotactics by now is 46 (ongoing and not final status). According these rules we can generate in the nominal paradigm from one noun root 112 word forms. Full details concerning the Kazakh nominal paradigm can be found in (Kairakbay, 2013).

3 Finite State Transducer for the Kazakh Nominal Paradigm

Creating a morphological analyzer/generator is based on the nominal paradigm of the Kazakh language as applied to the noun. For the formation the finite state transducer we used XFST (Xerox Finite State Tools) (Beesley and Karttunen, 2003).

Properly a process of building up morphological analyzer consists of the following steps:

- Noun morphotactics description;
- Morphophonemics rules description;
- Finite state automata (transducer) network formation.

Let’s consider these steps more in details.

3.1 Noun morphotactics description

Morphotactics description is carried out using a special language lexc. lexc is high-level and declarative programming language. The finite state automata formation is done by a special compiler lexc (Lexicon Compiler). Affix morphemes are designated with so-called Multichar Symbols. These symbols need to be described at the beginning of the file after Multichar_Symbols declaration. The following is an example of declaring Multichar Symbols:

```
Multichar_Symbols
+N              ! Noun
+Pl             ! Plural
+Sngl           ! Singular
+Poss12Sngl     ! Possession 1-2:Singular
+Poss12Pl       ! Possession 1-2:Plural
```

After Multichar Symbols declaration lexc program body is described. The body consists of LEXICONs. LEXICON is one of morpheme composing a word. At the beginning LEXICON Root must be declared. It corresponds to the Start State of the resulting Network. There can be declared roots of words if file is formed for one part of speech or we can declare part of speech if network is building up for the whole language. After LEXICON Root all remaining morphemes are described according the morphotactics rules. Next is an example of lexc file body for the Kazakh nouns: *ǻke – father* (English), *tôlem – payment* (English).

```
LEXICON Root
        Noun;
LEXICON Noun
        ǻke             NounTag;
        tôlem           NounTag;
        LEXICON NounTag
%+N:0                           SingularPlural;
LEXICON SingularPlural
%+Pl:lAr/DAr                    #;
%+Sngl:0                        #;
```

This is the example of description of noun partial morphotactics. In LEXICON Root a part of speech is declared – noun as Noun. Then LEXICON Root which contains word roots (ǻke, tôlem, etc.) is necessary to describe, and to point out transition to the next morpheme NounTag. LEXICON NounTag does not contain any morpheme if we point out null in expression %+N:0 but it adds Multichar Symbol +N which allows us to identify the word as noun at morphological analysis. % sign shields the + sign to use it only as a symbol, not lexc-operator because else the compilator will generate an error in given case. Further next morpheme SingularPlural is pointed out in LEXICON NounTag. In LEXICON SingularPlural we add plural morpheme lAr/DAr (formalized denotation lAr/DAr after morphophonemic rules is transformed to one of final endings: lar/ler, dar/der, tar/ter) and respective Multichar Symbol: +Pl. Symbol # indicates end of the word. Recall that this example is only a small part of the whole and does not describe a noun morphotactics for the Kazakh language. As a whole the description of all noun morphotactics is carried in a like manner.

3.2 Morphophonemics rules description

Morphophonemic rules are described in XFST by regular expressions and replace rules. Let's present a general scheme of replacement rule: `upper -> lower || left _ right`, where upper, lower, left, and right are regular expressions designating regular languages.

Example for Kazakh: a -> e || [g | k | ņ] _ [g | k | ņ] which can be read in natural language as "character 'a' replaces onto character 'e' if one of the letters [g | k | ņ] is located before it, and of the letters [g | k | ņ] is located after it". Here is an example of morphophonemics rules writing for the plural affix.

```
define plural1f  [ {lAr/DAr} -> {der} || FrontStem
[LMNN | JZ] _ ];
define plural1b [ {lAr/DAr} -> {dar} || BackStem
[LMNN | JZ] _ ];
```

where plural1f, plural1b, plural2f, plural2b, plural3f, plural3b are declared names of replace rules (name can be any suitable word consisting of Latin letters and numbers); FrontStem, BackStem, LMNN, JZ are the regular expressions.

```
define Consonants [ b | v | g | ġ | d | ž | z | j | k | ķ | l |
m | n | ņ | p | r | s | t | f | h | ḥ | c | č | š | ŝ | " | ‘ ]; #
Expression of consonants
define BackVowels [ a | o | u̇ | y | ë | û | â | u | i ]; #
Definition of back vowels
define FrontVowels [ǻ | ô | ù | ì | e | u | i ]; #
Definition of front vowels
define BackStem [ Consonants* BackVowels+
Consonants* ]; # Definition of back syllable
define FrontStem [ Consonants* FrontVowels+
Consonants* ]; # Definition of front syllable
define JZ [ ž | z ]; # Definition of letters ž or z
define LMNN [ l | m | n | ņ ]; # Definition of letters l,
m, n, and ņ
```

Let's analyze one of the rule: [{lAr/DAr} -> {der} || FrontStem [LMNN | JZ] _];

This rule can be interpreted as "Replace {lAr/DAr} onto {der} if preceding front syllable is ending on one of characters [l m n ņ ž z]". In a like manner we describe all morphophonemic rules including all exceptions.

3.3 Finite state automata (transducer) network formation

Once we have described morphotactics and the necessary rules of morphophonemics we need to compose them into a finite transducer network for the final analysis and generation of word forms. The joining up takes place by using the XFST instruments. After coupling of networks into transducer the following set of word forms is obtained:

Upper side:
ǻke+N+Pl ǻke+N+Sngl tôlem+N+Sngl tôlem+N+Pl
Lower side:
ǻkeler ǻke tôlem tôlemder

Simplified finite state representation of the Kazakh nominal paradigm is shown in fig.1.

4 Error correction

For very preliminary testing we chose 5 pages of text containing 1630 words of economic and business lexis. This text beforehand was processed by our OCR-module for the Kazakh language text recognition. Then we used the generated from Bektayev (1996) dictionary word forms by morphological module for the comparing with the words from OCR-processed text with the aim of search matching (edit distance equals 1). Results are given in the table 1.

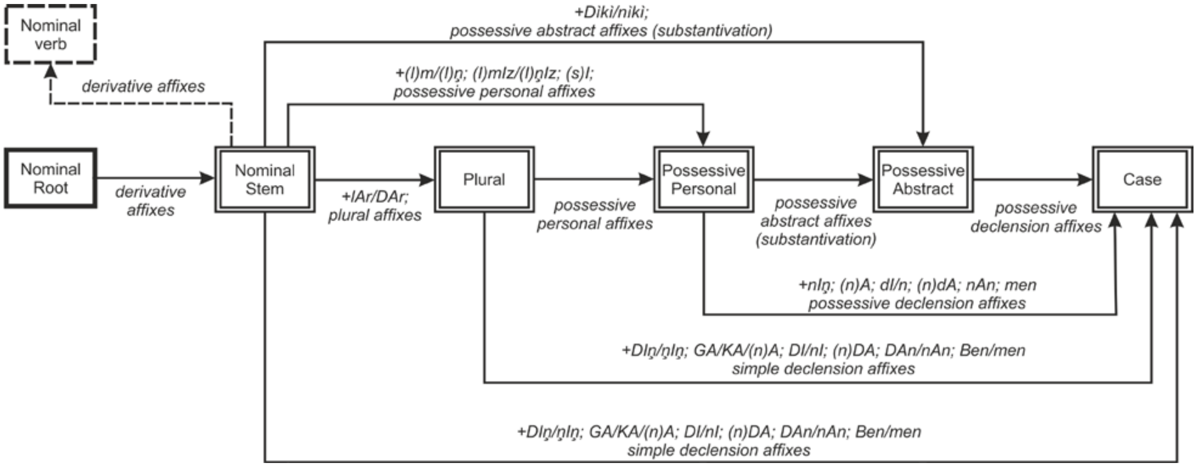
Nominal verb
derivative affixes
+Diki/niki;
possessive abstract affixes (substantivation)
+(l)m/(l)ŋ; (l)mlz/(l)ŋlz; (s)l;
possessive personal affixes
Nominal Root
derivative affixes
Nominal Stem
+lAr/DAr;
plural affixes
Plural
possessive personal affixes
Possessive Personal
possessive abstract affixes (substantivation)
Possessive Abstract
possessive declension affixes
Case
+nlŋ; (n)A; dl/n; (n)dA; nAn; men
possessive declension affixes
+Dlŋ/ŋlŋ; GA/KA/(n)A; Dl/nl; (n)DA; DAn/nAn; Ben/men
simple declension affixes
+Dlŋ/ŋlŋ; GA/KA/(n)A; Dl/nl; (n)DA; DAn/nAn; Ben/men
simple declension affixes

Figure 1. Simplified Finite State Representation of the Kazakh Nominal Paradigm

As seen using of constructed nominal paradigm allows to improve correction ratio of OCR-module in 1,67 times on average. Introduced errors (in last column of table) are connected with incompleteness of the Kazakh language paradigms formulation for another (than a noun) parts of speech. If we take into account only the number of corrected words which were caused exactly by formulated nominal paradigm then we get average 78%-level of correction. Further improvement can be achieved by the completeness of formulation of the Kazakh language paradigms, addition of specific professional lexicons, editing and cleaning up of dictionary base, and using error-tolerant algorithms of error correction (Oflazer, 1996).

5 Conclusion

We've presented in the paper the finite state approach to the Kazakh nominal paradigm. The main objective is to describe the formalized Kazakh nominal paradigm and to construct its finite state representation with the formation of correspondent finite state transducer that realizes morphological analysis/synthesis functions. We had a very preliminary testing on the use of morphological analyzer after OCR-processing module for correcting errors in the sample Kazakh text. Further the quality would be improved via the completeness of formulation of the Kazakh language paradigms, addition of specific professional lexicons, addition, editing and cleaning up of dictionary's database, and using error-tolerant algorithm of error correction.

File name	The number of words	The number of incorrect words after OCR-processing (% from the number of words)	The number of corrected words (% from number of errors)	The number of introduced errors (% from the number of words)
scan1.tif	315	31(10%)	29(94%)	23(7%)
scan2.tif	295	14(5%)	11(79%)	19(6%)
scan3.tif	293	21(7%)	17(81%)	17(6%)
scan4.tif	352	41(12%)	32(78%)	21(6%)
scan5.tif	375	50(13%)	33(66%)	18(5%)
In total	1630	157(10%)	122(78%)	98(6%)

Table 1. Preliminary Testing of Error Correction in Selected Text (Economic and Business Lexis)

Acknowledgments

We would like to thank the reviewers for valuable comments.

This work was supported by the grant from the Ministry of Education and Science of Republic of Kazakhstan.

References

Iñaki Alegria, Xabier Artola, Kepa Sarasola, and Miriam Urkia. 1996. Automatic morphological analysis of Basque. *Literary and Linguistic Computing*, 11(4):193-203.

Iñaki Alegria, Maxux Aranzabe, Nerea Ezeiza, Aitzol Ezeiza, and Ruben Urizar. 2002. Using Finite State Technology in Natural Language Processing of Basque. *Lecture Notes in Computer Science*, 2494:1-12.

Gulila Altenbek and Xiao-long Wang. 2010. Kazakh Segmentation System of Inflectional Affixes. *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing* (CLP2010), Beijing, China, p.183–190.

Kemal Altintas and Ilyas Cicekli. 2001. A Morphological Analyser for Crimean Tatar. *Proceedings of the 10th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'2001)*, North Cyprus, p.180-189.

Mohammed Attia, Pavel Pecina, Antonio Toral, Lamia Tounsi, and Josef van Genabith. 2011. An Open-Source Finite State Morphological Transducer for Modern Standard Arabic. *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, Blois, France, p. 125–133. Association for Computational Linguistics.

N. A. Baskakov. 1981. *Altaic Family of Languages and its Study*. Nauka publishers, Moscow, Russia. (in Russian).

N. A. Baskakov, A.K. Khasenova, V.A. Issengaliyeva, and T.R. Kordabayev (eds.). 1966. *A comparative grammar of the Russian and Kazakh languages. Morphology*. Nauka publishers, Alma-Ata, Kazakhstan. (in Russian).

Kenneth R. Beesley. 2001. Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. *Proceedings of the Arabic Language Processing: Status and Prospect--39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France.

Kenneth R. Beesley and Lauri Karttunen.2003. *Finite State Morphology*. CSLI Publications, Stanford, CA.

Kaldybaj Bektayev. 1996. *Large Kazakh-Russian and Russian-Kazakh dictionary*. Kazakhstanskij proekt publishers, Almaty, Republic of Kazakhstan.

Çağrı Çöltekin. 2010. A Freely Available Morphological Analyzer for Turkish. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Gülşen Eryiğit and Eşref Adalı. 2004. An Affix Stripping Morphological Analyzer for Turkish. *Proceedings of the IASTED International Conference Artificial Intelligence and Applications*, Innsbruck, Austria.

ISO 9: 1995. Information and documentation. Transliteration of Cyrillic characters into Latin characters. Slavic and non-Slavic languages. *International Organization for Standardization*.

Bakyt M. Kairakbay, Ilyas E. Tursunov, and David L. Zaurbekov. 2012. A Design of Computer Recognition System of Kazakh Language Text: OCR, Morphotactics and Morphophonemics. *Proceedings of the 3rd World Conference on Information Technologies (WCIT 2012)/Elsevier Procedia Technology*, Barselona, Spain. (to be published).

Bakyt M. Kairakbay. 2013. A Nominal Paradigm of the Kazakh Language. *1st International Symposium "Morphology and its Interfaces"*, Lille, France. (submitted).

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and productio*n. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki, Finland.

Karl A. Krippes. 1996. *Kazakh Grammar with Affix List*. Dunwoody Press, Kensington, MD.

K.M. Mussayev 2008. *The Kazakh Language*. Vostochnaya literatura publishers, Moscow, Russia. (in Russian).

Kemal Oflazer. 1994. Two-level Description of Turkish Morphology. *Literary and Linguistic Computing*, 9(2):137-148.

Kemal Oflazer. 1996. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Journal of Computational Linguistics*, 22(1):73-89.

Murat Orhun, A.Cüneyd Tantug and Esref Adalı. 2009. Rule Based Analysis of the Uyghur Nouns. *International Journal on Asian Language Processing*, 19 (1): 33-43.

Cüneyd Tantuğ, Eşref Adalı, and Kemal Oflazer. 2006. Computer Analysis of the Turkmen Language Morphology. *Lecture Notes in Computer Science*, 4139:186-193.

Jonathan North Washington, Mirlan Ipasov, and Francis M. Tyers. 2012. A finite-state morphological transducer for Kyrgyz. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Aishan Wumaier, Parida Tursun, Zaokere Kadeer, Tuergen Yibulayin. 2009. Uyghur Noun Suffix Finite State Machine for Stemming. *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009)*, Beijing, China, p.161-164.