Linguistics and Literature Studies 4(1): 96-104, 2016
DOI: 10.13189/lls.2016.040111 http://www.hrpub.org

A Rule Based Morphological Analyzer and A Morphological Disambiguator for Kazakh Language

Gulshat Kessikbayeva, Ilyas Cicekli*

Department of Computer Engineering, Hacettepe University, Ankara, Turkey

Abstract Morphological analysis is a very critical issue especially for natural language processing related tasks on agglutinative languages. This study gives the implementation details of a rule-based morphological analyzer of Kazakh language which is an agglutinative language. A detailed computational analysis of Kazakh language morphology such as formalization of alternation and morphotactic rules for Kazakh language is worked out in order to create the morphological analyzer. In the implementation of the morphological analyzer, alternation and morphotactic rules of Kazakh language are represented by two-level morphology rules and Foma finite state compiler is employed. This is the first detailed computational analysis of Kazakh language from morphological view. A word can have more than one morphological parse but only one of its morphological parses is valid in a given sentence. A morphological disambiguator disambiguates words by selecting one of possible parses of words. In this paper, we also present a transformation-based morphological disambiguator for Kazakh language and it is a variation of Brill tagger.

Keywords Morphological Analysis, Morphological Disambiguation, Natural Language Processing.

1 Introduction

Kazakh Language is a Turkic language which belongs to Kipchak branch of Ural-Altaic language family, and it is spoken approximately by 8 million people. It is the official language of Kazakhstan and it has also speakers in many countries. It is closely related to other Turkic languages such as Turkish and there exist mutual intelligibility among them. Although different word orders are possible for Kazakh sentences, the main word order is verb-final order same as Turkish.

Kazakh language is an agglutinative language and Kazakh words can be generated from root words recursively by adding proper suffixes representing morphemes. From a single root word, too many words can be generated using derivational and inflectional morphemes. The order of added morphemes are governed by morphotactic rules of the language. A same morpheme can be realized as different suffixes depending on letters of root words. Surface level realizations of these morphemes are governed by the root word vowel harmony property of the language. Although most of Kazakh words obey the vowel harmony property, there are some loan words that do not obey this property. Most of these loan words come from other languages such as Russian, Persian and Arabic.

Many natural language processing (NLP) related tasks on agglutinative languages require morphological analysis step since sentence structures and meanings are governed by morphological structures of words. The meaning and grammatical role of a word in a sentence can be obtained from the morphological structure of that word. Thus, having a morphological analyzer is a starting point for many NLP related researches. Words can have more than one morphological parse and this causes morphological level ambiguity in natural languages. Although a word can have more than one morphological parse, only one of its morphological parses is intended in a given sentence. A morphological disambiguator tries to find intended morphological parses of words in sentences.

Generally a morphological analyzer is built as a finite state transducer (FST) based on a formal description of the morphology of that language. Morphological analysis can be considered as a finite state process and there are many other successful applications of finite state techniques in various areas of NLP [21]. In natural language, a word can be a root word or created from a root word by affixing possible morphemes to that root. Thus, a word taken into a FST is checked for all possible root words and possible morphemes affixed to those root words. The FST representing the morphological analyzer returns all possible morphological parses of given words. Morphology level ambiguity can be handled by using a morphological disambiguator.

Finite state environment tools such as Foma [11] can create a rule-based morphological analyzer for a natural language from its two-level morphology rules that represent alternation and morphotactic rules of that language. In order to create a rule-based morphological analyzer for a language, two sets of two-level morphology rules should be created which describe the morphology of that language. The first set is the set of orthographic rules describing spelling and alternation rules of that language. The second set is the set of morphotactic rules that describe the order of morphemes in words. We created orthographic and morphotactic rules for Kazakh language and created a rule-based Kazakh morphological analyzer from these rules using Foma finite state environment tools.

This paper gives a deep analysis of Kazakh language morphology by creating a rule-based morphological processor for Kazakh language. The finite state transducer representing this morphological processor is created from the developed two-level morphology rules. The morphological processor can work in both direction such that it can analyze the surface level representation of a given word in order to find its possible lexical level representations and it can generate its surface level of the given word from its lexical level representation. The surface level representation of a word is its normal usage in the language and the lexical level representation indicates morphemes of that word. Two-level morphology is a way of handling morphological structures by executing parallel rules [3].

In order to produce a morphological analyzer, there are also some statistical based or data driven approaches which do not require deep language analysis and they are treated as lightweight morphological analyzers. A deep analysis of the morphology of the language is essential for further less error prone works. Due to its agglutinative property of Kazakh language where every affix converts a given word to a different form, a word can have many different morphological parses. Even a word can have different parses with a same part of speech. On the other hand, a lightweight analysis of the language has to deal with more errors in next stage. For this reason, we preferred the creation of a rule-based morphological analyzer for Kazakh language with a deep analysis of its morphology.

Since words can have many morphological parses, this causes ambiguity problem at morphology level. Although a word can have different parses, only one of its parses is intended for a given sentence. We also developed a morphological disambiguator for Kazakh language in order to select intended parses of words. The developed disambiguator is transformation-based rule morphological disambiguator and it is a variation of Brill tagger [5].

The paper is organized as follows. In Section 2, an overview of related work is given. Section 3 gives a brief overview of the Kazakh writing system and script and detailed information about vowel and consonant harmony rules. Then, the inflectional system for nouns is presented and morphotactic rules of nominal roots, pronouns, adjectives, adverbs and numerals are explained in Section 4. After that the detailed analysis of verbs and verb tenses are introduced in Section 5. The morphological disambiguation system which has been worked out for Kazakh is described in Section 6 ; evaluation results and their analysis are discussed in Section 7. Finally, conclusions and future work are presented in Section 8.

2 Related Work

There are many works performed on working out morphologies of natural language. These works can be classified as rule based, statistical or data driven methods and hybrid methods. Rule based morphological analyzers with FSTs for many languages including Finnish, Swedish, Russian, English, Swahili and Arabic have been developed [16]. Moreover, many studies and researches have been done upon on morphological analysis of Turkic Languages. The morphological analysis of Turkish is performed by a Turkish morphological processor developed earlier [13] which uses morphology rules defined by Oflazer [23]. Affix types and grammatical names in Kazakh morphological processor worked out in this paper are also defined similarly to Turkish morphological processor [23, 13]. There is a rule-based morphology analysis of Crimean Tatar developed for translation system which involves Turkish to Crimean Tatar in 2001 by Altintas and Cicekli [2]. Moreover, there is a morphological analyzer of Turkmen language worked out by Tantug [29]. In addition a rule-based morphological analysis of Uygur was developed by Orhun [26]. Also, a freely available Morphological Analyzer for Turkish is proposed by Cagri [6].

Especially for Kazakh language there is a considerable increase in NLP related research areas. Analysis of inflectional affix of Kazakh Language was studied within the work of Kazakh segmentation system [1]. A finite state approach for Kazakh nominals are presented by Kairakbay [14]. This paper studies rules of alternations specific for each case, rather than generalized form. It can bring to over loaded size of rules for all grammar. Washington et al. developed Finite-state morphological transducer for three Kypchak languages [33] including morphology for Kazakh language with limited stem size in lexicon. Also, Mahambetov et al. worked on Kazakh morphology with data-driven method by evaluating on the large data set with 97% accuracy while certain language-specific issues are not considered.

Our rule based morphological processor for Kazakh language differs from above works in that: First, it gives deep analysis of a language with inflectional and derivational morphemes. Also, it covers nearly all language-specific issues. Finally, it does not require huge word-based data sets of Kazakh language for morphological processor. The coverage of our morphological analyzer is substantial and its accuracy is 99%. It only does not cover some loan words, technical words and proper nouns.

Morphological disambiguators can be categorized as statistical, rule-based and hybrid systems. Statistical methods [7, 28] create probabilistic models from morphologically tagged texts and use these models to disambiguate words by selecting most probable morphological parses. There is a statistical morphological analysis for Turkish worked out using n-gram models for inflectional and final tags of words [10]. Rule-based morphological disambiguators [24, 25, 8] use hand-crafted rules to select correct parses of words or eliminate some of illegal parses of words. Disambiguation rules can be also learned from tagged texts using transformation-based learning approaches [5]. Hybrid systems [30] use both statistical knowledge and disambiguation rules in disambiguation process. Turkish morphological disambiguator developed [18] by Kutlu and Cicekli uses both transformation-based approach and rule-based approach. The morphological disambiguator for Kazakh language described in this paper use transformation-based approach and it is a variation of Brill tagger [5].

3 Vowel and Consonant Harmony

Kazakh is officially written in Cyrillic alphabet. In its history, it is represented by Arabic, Latin and Cyrillic letters. Nowadays switching back to Latin alphabet in 20 years is planned by Kazakh government [27]. In this paper, the current Cyril version is used for convenience.

Two main issues of language such as morphotactics and alternations can be dealt with finite-state tools. In our morphological analyzer, morphotactic rules are represented by encoding a finite-state network and a finite-state transducer for alternations is constructed using Foma finite-state tools [11]. Then, the formed network and the transducer are composed into a single final network which cover all morphological aspects of the language such as morphemes, derivations, inflections, alternations and geminations [4].

Vowel harmony of Kazakh language obeys the following rule: vowels in each syllable should match according to being front or back vowel. It is called synharmonism and it is basic linguistic structure of nearly all Turkic languages [9]. For example, the word қала-лар-дың ("of cities") has the stem қа-ла ("city") whose two syllables contain back vowels and all added suffixes should contain back vowels according to the vowel harmony rule. Both of its suffixes –лар and –дың contain back vowels. Here, –лар is an affix of plural form and –дың is an affix of genitive case. However, as stated before, there are a lot of loan words from Persian and generally they do not obey the vowel harmony rule. For example, мұ-ға-лім means "teacher", and its first two syllables have back vowels and its last syllable has a front vowel. Since suffixes to be added are defined according to the last syllable, the vowel of the last syllable should match with all other remaining morphemes. For example, the word мұғалім-дер-дің ("of teachers") whose last two syllables contain front vowels obeys the vowel harmony rule. On the other hand, there are morphemes with static front vowels which regardless from the type of the last syllable can be added to all words such as instrumental suffix –мен which contains a front vowel. In this case, all suffixes added after that suffix should also contain front vowels. Words in Kazakh language take suffixes with vowels а or ы if their last syllables contain back vowels, and in other cases they take suffixes containing vowels е or і.

The developed morphological analyzer provides mappings between lexical and surface level representations of Kazakh words. Although users only deal with lexical and surface level representations of the words, the morphological analyzer also uses intermediate representations of words. In order to construct a finite state transducer for alternation rules, some capital letters such as A, J, H, B, P, C, D, Q, K are defined at intermediate level and they are invisible by users. These representations are used for substitutions such as A is for а and е and J is for ы and і. So, if the suffix дA should be added to a word according to morphotactic rules, it means that actual suffixes да or де are considered in accordance with alternation rules. There are groups of letters that are defined according to their sounds and these groups are used in alternation rules [32].

Consonant harmony rules are varied according to last letters of words in morphotactic rules. As in Table 1, different patterns are presented in order to visualize the relation between common valid rules and generalize morphotactic rules. Consonants in Table 1 are divided into three groups such as sonorous, voiced and unvoiced consonants. Sonorous and voiced consonants are also grouped as Type 1 and Type 2. In Table 1, Type 2 unvoiced consonants and unvoiced consonants have same pattern and this means that similar suffixes are added after them. Thus, Table 1 defines five different patterns which affect suffix types to be added to words according to morphotactic rules.

Table 1. Groups of Kazakh letters according to their sound (GLS)

Name	Type 1	Type 2
Sonorous Consonant	л р у й	м н ң
Voiced Consonant	з ж	б в г ғ
Unvoiced Consonant	п ф қ к т с ш щ х ц	
Vowel	а е ә і ы ө о ұ ү и	

Table 2. First Group of Similar Alternation Rules according to GLS

GROUP 1					
Ablative Case		Locative case		Dative Case	
дAн		дA		ТA	
тAн		тA		ТA	
нAн	3	ндA	3	нA	3
				A	1/2

Table 3. Second Group of Similar Alternation Rules according to GLS

GROUP 2					
Genitive Case		Accusative case		Poss. Affix-2	
дJң		дJ		дікі	
тJң		тJ		тікі	
нJң	3	нJ		нікі	
		н	3		

All rules for suffixes depend on last letters of morphemes in morphotactic rules. Table 2 and Table 3 give some groupings that can be made in order to set some generalized rules overall. Patterns of last letters of morphemes in Table 2 and Table 3 are matched with groups of letters presented in Table 1. For example, locative case affix is –дA, if the last letter is vowel, sonorous consonant or voiced consonant of Type 1. It is –тA, if the last letter is unvoiced consonant or voiced consonant of Type 2. It is –ндA, if the last letter is ы or і, since a word is at third personal possessive state. Here A is for a or e according to the last syllable of containing front or back vowel. So visually some cases have similar patterns and some are exactly the same [32]. Here boxes presented by numbers such as 1, 2 and 3 are for personal possessive agreements.

For example, word әке ("father") in ablative case with none possessive agreement will take suffix –ден, but in third person possessive agreement it takes suffix –нен. Thus әке+Noun+A3Sg+Pnon+Abl→әке–ден ("from father") and әке+Noun+A3Sg+P3Sg+Abl→ әке–сі–нен ("from his father") mappings occur. This is different from words which end with vowels. For example, a word сіңлі means "little sister" and its ablative case is analyzed as сіңлі+Noun+A3Sg+Pnon+Abl→сіңліден. According to those similarities there are generalized rules which are valid in many cases in grammar including verbs and derivations.

In Table 2, locative and dative suffix rules are nearly identical which can be observed visually. Also, accusative and possessive pronouns of Type 2 are the same. In dative case, if the last letter is a vowel and the last syllable contains a back vowel then T is replaced by ғ or г. Also, if the last letter is an unvoiced consonant and the last syllable contains a front vowel then T is for letters қ or к. Thus, the word бала ("child") becomes бала–ға ("to child") and the word әке ("father") will be әке–ге ("to father") in ablative case. The reader can observe that the last letter is a vowel, at the same time it is a front vowel in the last syllable and thus T→г mapping occurs. Also, the last letter of the word кітап–қа ("to book") is an unvoiced consonant and its last syllable contains a back vowel, thus T→қ mapping occurs. The last letter of the word мектепке ("to school") is an unvoiced consonant and its last syllable contains a front vowel, thus T→к mapping occurs.

After detailed analysis of the language it can be seen that there are mainly common rules of alternations valid over all grammar. There are about 57 main alternation rules defined for all system together with generalized rules and 13 exception rules for each case separately. All these rules are implemented with Foma finite-state tools, and they are defined and composed in a Foma file [11]. For instance, some of most common alternation rules are given below and they are called by capital letters defined at intermediate level and they are not accessible at surface level. As mentioned before they are invisible by users. They are represented by surface level characters or they drop. In the following rules, 0 stands for empty string.

Rule H & Rule B: H is realized as 0 or J, B is realized as 0 or A.

$[H \rightarrow 0, B \rightarrow 0 \mid\mid [Vowel]\% + \_[Cons]]$
$[H->J, B->A]$

If the last letter of the morpheme is a vowel and the first letter of the suffix is a consonant then H and B are realized as 0. Otherwise, they are realized as J and B, respectively. Some examples of Rule H and Rule B are as follows, and two of examples also uses Rule J and Rule A.

ана-Нм→ана-м, "my mother"
іш-Нм→іш-Jм→іш-ім with Rule J, "my stomache"
еге-Вр→еге-р, "will sharpen"
бар-Вр→бар-Ар→бар-ар with Rule A, "will go"

Rule J & Rule A: J is realized as ы or і and A is realized as й, а or е.

$[A \rightarrow й \mid\mid [Vowel]\% + \_]$
$[A \rightarrow а, J \rightarrow ы \mid\mid [BVowel](Cons) * \%+? * \_]$
$[A \rightarrow е, J \rightarrow і \mid\mid [FVowel](Cons) * \%+? * \_]$

If the last letter of morpheme is a vowel then A is realized as й, and if the last syllable of a morpheme contains a back vowel then A and J are realized as а and ы. Otherwise, if the last syllable of a morpheme contains a front vowel then A and J are realized as е and і. Some examples of Rule R and Rule A are as follows.

бас-Нм→бас-Jм→басым, "my head"
дос-тАр→дос-тар, "friends"
дәптер-лАр→дәптер-лер, "copybooks"
барма-Амын→барма-ймын, "I will not go"

Rule T: T is realized as қ, ғ, к or г depending on previous characters.

$[T \rightarrow қ \mid\mid [BVowel](?)[UVCons]\% + \_]$
$[T \rightarrow к \mid\mid [FVowel](?)[UVCons]\% + \_]$
$[T \rightarrow ғ \mid\mid [BVowel](?)[0|SCons|VCons1]\% + \_]$
$[T \rightarrow г \mid\mid [FVowel](?)[0|SCons|VCons1]\% + \_]$

This rule is illustrated partly in Table 2 for dative case. It is one of generalized rules which are valid in many cases such as derivation of nouns, adjectives and verbs. Some examples of Rule T are as follows.

бала-Та→бала-ға, "to child" (Noun in Dative)
жаз-Ты→жаз-ғы, "of summer" (Adjective)
жүр-Телі→жүр-гелі, "since coming" (Verb)
естіт-Тіз→естіт-кіз, "make hear"(Causative Verb)

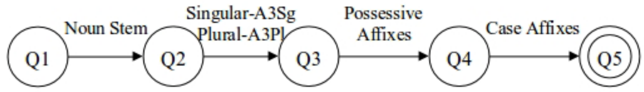

Figure 1. The FSA model of inflectional changes of a noun.

4 Nouns

Nouns in Kazakh language take singular or plural (A3sg, A3pl) suffixes, possessive suffixes, case suffixes and derivation suffixes. In addition, nouns can take personal agreement suffixes when they are derived into verbs. For example, мен мұғалім-мін which means "I am a teacher" has the following morphological analysis.

мен+Pron+PersP+A1Sg+Pnon+Nom
мұғалім+Noun+A3sg+Pnon+Nom
^DB+Verb+Zero+Pres+A1sg.

Every nominal root has at least a lexical form of Noun+Sg+Pnon+Nom. Therefore, a noun root кітап which means "book" has a morphological analysis as кітап+Noun+A3Sg+Pnon+Nom. These inflections of noun are given in FST diagram in Figure 1.

It can be seen that a nominal root can be in singular form by adding (+0) no suffix which is in fact third personal singular agreement (A3sg) and by adding suffix (+PAr) in plural form which is in fact third personal

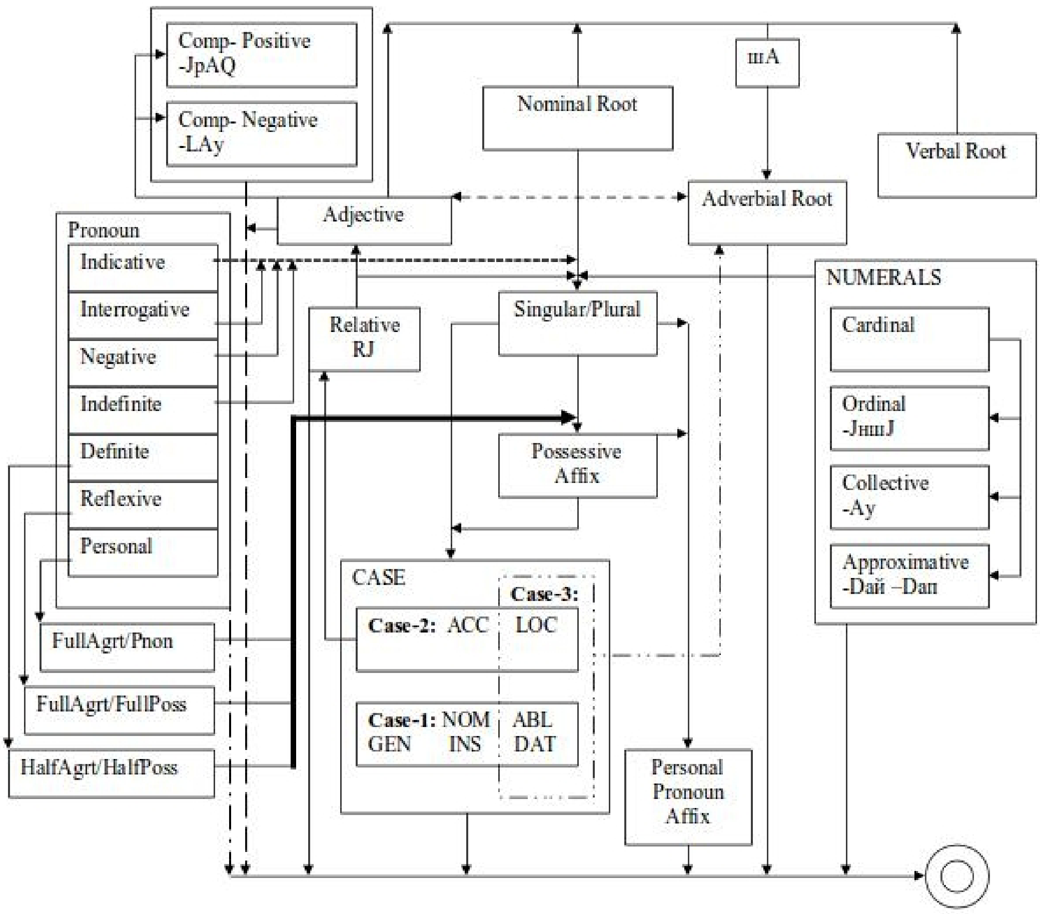

Figure 2. Morphotactic Rules for Nominal Roots.

plural agreement (A3pl). Here P is an intermediate level representation letter for letters д, т or л in surface level. After, possessive affixes (+Pnon:0, +P1sg:Hм, +P2sg:Hң, +P2psg:HңJз, +P3sg:cJ, +P1pl:HмJз, +P2pl:Hң, +P2ppl:HңJз, +P3pl:cJ) and case affixes (Nom, Dat, Abl, Loc, Acc, Gen, Ins) are added. Here H and J are intermediate letters. All morphotactic rules together with adjective, pronoun, adverb and numerals are given in Figure 2. It can be observed that every adjective can be derived to noun and nouns with relative affix can be derived to adjectives. There are other derivations which are produced by adding some specific suffixes between verbs and nouns, adjectives and adverbs, adjectives and nouns. In order to get rid of complex view those derivations are not explicitly shown in Figure 2. In our morphological analysis system, root of word is a starting point for morphemes defined in lexicon file, and other morphemes are added according to morphotactic rules. All possible morphemes of Kazakh language are defined in the lexicon of the morphological analyzer.

5 Verbs

Verbs are terms which define actions and states. Mainly three tenses exist such as present, future and past as stated in Figure 3. Moreover, conditional, optative and imperative moods are also defined. However in detailed form there are thirteen tenses together with modals in Kazakh language. These tenses are worked out from many resources where presentation and naming have variance among each other according to their scholars [12, 20, 22, 31]. For example, according to Isaeva and Nurkina [12] Ауыспалы Келер Шақ, “Future Transitional Tense” denotes action in future and has same affix as Present Tense. However, Mamanov [20] points out that Ауыспалы Келер Шақ, “Future Transitional Tense” denotes present action. Our work is mainly based on morphology of Kazakh language defined by Karaev in[15]. Additionally, there is large amount of auxiliary verbs which define tenses and some modal verbs. However, in cases that auxiliary verbs are not used as verbs, they become adverbial adverbs or participles which define verb or noun [9]. In Figure 4, morphotactic rules of verbs and modals are given. Derivations of verbs to nouns and adverbs with specific suffixes are shown with asterisk in Figure 4.

Verbs can be in reflexive, passive, collective and causative forms. For instance, verb тара-у which means "to comb" is represented as тара-н-у in reflexive infinity form, тара-л-у in passive infinity form, тара-с-у in collective infinity and тара-тQJз-у and тара-тTJр-у in causative infinity form. Here, Q, J and T are intermediate letters. However not all verbs can have all of these forms at the same time.

Verbs in infinity form are generally formed with last letter у, and the verb келу which means “to come” is an example for this case. The system is performing over generalization on verbs which take auxiliary verbs on appropriate tenses. Those verbs are analyzed as derived adverbs or incomplete verbs on that tense since every

	Шак/Tense	Жұрнак/Suffix	
Future - Келер	Болжамды Келер Шақ *Future Indefinite Tense*	ар/ер/р	
Future - Келер	Мақсатты Келер Шақ *Future Goal Oriented Tense*	бақ/бек пақ/пек мақ/мек	
Future - Келер	Ауыспалы Келер (Осы) Шақ *Future (Present) Transitional Tense*	а/е/й	*Present*-Осы
	Мақсатты Осы Шақ *Present Goal Oriented Tense*	қалы/ғалы келі/гелі	*Present*-Осы
	Нақ Осы Шақ *Present Definite Tense*	ып/іп/п	*Present*-Осы
	Дәл Осы Шақ *Present Progressive*	да/де	*Present*-Осы
Past - Өткен	Жедел Өткен Шақ *Past Definite Tense*	ды/ді ты/ті	
Past - Өткен	Ауыспалы Өткен Шақ Past *Transitional Tense*	атын/етін йтын/йтін	
Past - Өткен	Айғақты Бұрыңғы Өткен Шақ *Past Narrative Definite Tense*	ған/ген қан/кен	
Past - Өткен	Айғақсыз Бұрыңғы Өткен Шақ *Past Narrative Indefinite Tense*	ып/іп/п	

Figure 3. Tenses of Verbs in Kazakh Language.

verb of a sentence should have a personal agreement. It means the personal agreement affix is added to the verb itself after the tense suffix or to the auxiliary verb. Some of the tenses have different personal agreement endings and they are presented in Figure 4.

In the constructed morphological analyzer, we make the analysis of every single word and for that reason generalization of some rules is made by giving more than one result. Thus compound verbs are examined separately. For example, кел-гелі тұр-мын which means "I am planning to come" is an example of this usage. Here тұр is an auxiliary verb which actually defines the tense of the verb and takes a personal agreement affix. Without an auxiliary verb, the word кел-гелі means "since coming" and it is derived as an adverb. Thus, in order to choose a correct one we developed the disambiguation system which is explained in next section.

6 Morphological Disambiguation

Natural language is a complex issue due to its being natural and having mental influence of a speaker with effects of cultural, social, historical and geographical background of his society. Regardless from the context where it is used, a word in a natural language can have more than one meaning. This case is called the ambiguity of a word and it is a big issue to be considered for any natural language processing task with even well-defined grammar rules. Especially this ambiguity problem is more complex for agglutinative languages. Kazakh language is an agglutinative language in which every affix converts a given word to a different form. Thus, its morphological disambiguation process is harder than others because it has more morphological parses for words.

The morphological disambiguation system for Kazakh language is constructed using a variation of Brill Tagger [5]. Brill Tagger can be briefly summarized as an error-driven transformation-based tagger method which aims to minimize the total error. Our disambiguation system which is a variation of Brill Tagger is based on the idea of Kutlu and Cicekli [18], which was constructed for Turkish language earlier.

Our system consists of two main parts such as training and disambiguation processes. First of all, we created a corpus for morphological disambiguation and words of this corpus are analyzed using our morphological analyzer. The correct morphological parses of words are manually tagged. As a result, we obtained a manually tagged training corpus which has 30,171 words and it is used for training. We also created another test corpus which has approximately 15,000 words and it is used for validation purpose.

The training corpus is used to construct tables such as Most Likely Tag of Word Table (WTBL) and Most Likely Tag of Suffix Table (STBL). All morphological parse frequencies of words are kept in the table (WTBL) and all morphological parse frequencies of suffixes are present in the table (STBL) in sorted order. It means that the first tag for a word or a suffix has the highest frequency and thus it is the most likely tag in each case. Here morphological parse or tag of a word is taken as whole tag of a word.

A morphological parse of a Kazakh word can contain derivational and inflectional suffixes same as a Turkish word. A derivational suffix is shown by ^DB in lexical forms and it indicates a derivation boundary. Except for the stem of a word, its all other morphemes in its morphological parse is called the whole tag of that word. The collection of final morphemes after the last derivation is called as the final tag of the word. For example, morphological parse of қойшы (shepherd) is as follows.

қой+Noun+A3sg+Pnon+Nom^DB+Noun+A3sg +Pnon+Nom.

Here, the whole tag of the word қойшы is

Noun+A3sg+Pnon+Nom
^DB+Noun+A3sg+Pnon+Nom

and the final tag is "Noun+A3sg+Pnon+Nom". If a word doesn't have any derivation boundaries its whole tag is its final tag.

At this stage, disambiguation rules are induced by using tables (WTBL, STBL). In our disambiguation system, the induced possible rules are in the following 3 forms.

- Type1: Select TAG_A for $WORD_N$ if the tag of $WORD_{N-1}$ is TAG_B.
- Type2: Select TAG_A for $WORD_N$ if the tag of $WORD_{N-1}$ is TAG_B and if the tag of $WORD_{N+1}$ is TAG_C
- Type3: Select TAG_A for $WORD_N$ if the tag of $WORD_{N+1}$ is TAG_C, where TAG_A, TAG_B

and TAG_C are possible tags from WTBL. Here "Select TAG_A for $WORD_N$ if Condition" means that we select the morphological parse with TAG_A for $WORD_N$ if "Condition" is satisfied and TAG_A is the tag of at least one of the morphological parses of $WORD_N$. If there is more than one morphological parses with TAG_A which belongs to that word, select the one with the highest frequency. If $WORD_N$ does not have a morphological

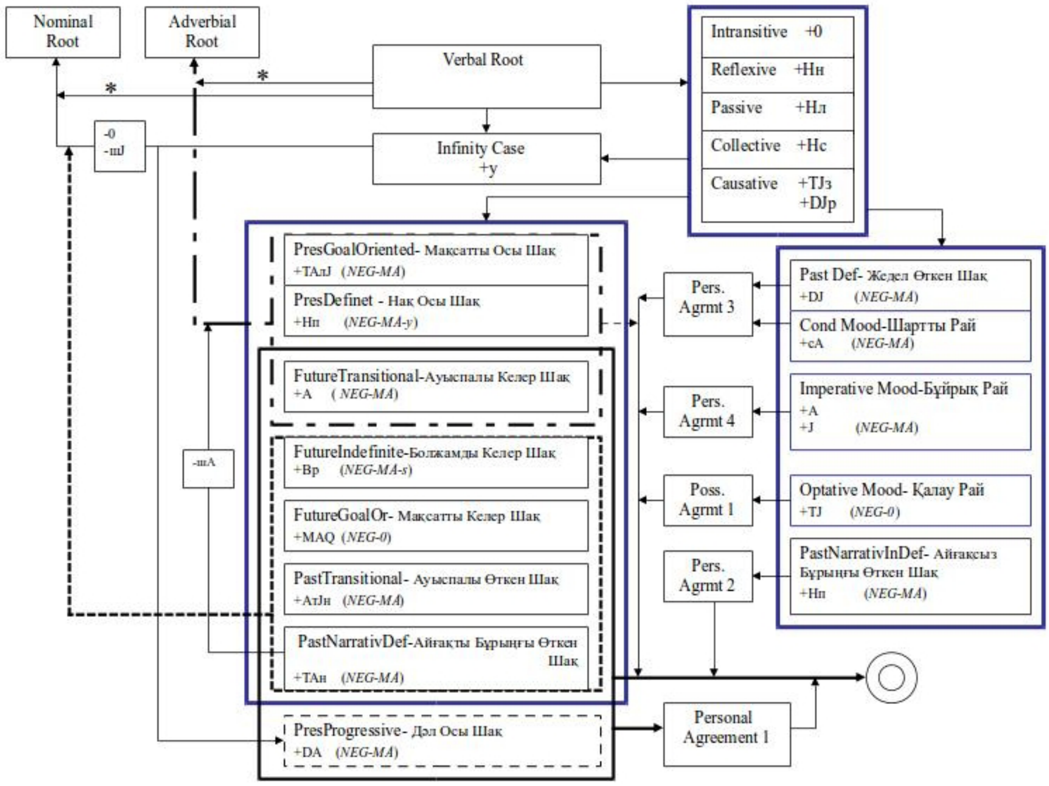

Figure 4. Morphotactic Rules of Verbs in Kazakh Language.

parse with TAG_A, the rule does not have any effect on $WORD_N$.

After all possible rules are found, each rule is tried in order to select the rule that gives the best precision-increase. Here the precision value is evaluated as follows:

$$Precision = \frac{\text{Number of Correctly Tagged Words}}{\text{Number of Total Words}}$$

where Number of Correctly Tagged Words is the number of correctly tagged words in the data set (here the data set is the training set with most likely morphological parses), and Number of Total Words is the number of the words in the data set. After applying the selected rule, we repeat the process until there is no progress or the improvement after the last found best precision. All learned rules are kept in their learning order. Then, WTBL, STBL and the learned rules are used in the disambiguation process.

The disambiguation system consists of four major components such as:

- Selection of Most Likely Tag of Word
- Selection of Most Likely Tag of Suffix
- Selection Most Likely Tag with Fall-Back Heuristics
- Application of Learned Rules

The system looks for the correct morphological parses applying the above components in the given order. The most likely tag of each word is selected with one of the three four components. After the selection of most likely tags for words, the learned disambiguation rules are applied to find correct parses of words.

If a word is available in WTBL, the mostly tag in WTBL is selected for that word. Otherwise, STBL is checked whether the suffix of that word is available in STBL. If its suffix available in STBL, the most likely tag of the suffix is selected as most likely tag of that word. Certainly, we can not have all words in our training corpus and some words can be still ambiguous after first two steps have been applied. In this case, the third step which is "Selection with Fall-Back Heuristics" will force the system definitely select a parse for each ambiguous word. Differently from the disambiguation system [18], if word is unknown we try to find a word by chunking a word from the last letter to find valid previously learned suffixes. For example, assume a word “сатып” which means "selling" is ambiguous. We look for the last letter, which is ”п” as suffix and the rest word, which is ”сатығ” as a stem. If we have such predefined suffix in STBL, we will take all most frequent parses. On the other side, we look at a stem in WTBL. We are continuing this process until a stem with one letter is left. There is a possibility of having an unknown word without any predefined suffix. In this case, it is assumed that this unknown word has for possible morphological parses such as a noun, an adjective derived from noun, a verb derived from noun and an adverb derived from adjective. It is also assumed that its most likely tag is noun.

7 Tests and Analysis

As mentioned before, the system is implemented using Foma finite state tools [11]. Morphotactic rules and possible morphemes are defined in the lexicon file. Alternation rules of Kazakh language are defined and the rules are composed with the lexicon file in a Foma file. Some loan words, proper names and technical terms are not included. The system is working in two directions as at lexical and surface level. Due to the ambiguities in language there is no one-to-one mapping between surface and lexical forms of words and the system can produce more than one result.

There are approximately 15000 words in our test corpus which are selected from the web [27]. The percentage of correctly analyzed words is approximately %99. In the lexicon of the morphological analyzer, there are 3709 verbs, 13149 nouns, 3047 adjectives, 1218 adverbs, 794 conjunctions and 100 postpositions and numerals are included.

Table 4. Test Results of Morphological Analyzer

files	total words	correct	uncorrect	average parse per word	precision
1.txt	6462	6432	30	7.09	0.995
2.txt	3124	3093	31	6.91	0.990
3.txt	2836	2784	52	7.11	0.982
4.txt	2532	2493	39	6.65	0.985
Total	14954	14802	152	6.98	0.990

The errors of the morphological analyzer are mainly the errors that appear in the analysis of technical, abbreviated and loan words which do not obey alternation rules of Kazakh language. The system is tested with four files in our test corpus and their results are given in Table 4. The files 1.txt and 2.txt have less such words than the files 3.txt and 4.txt. For example, the word фактілер which means "facts" is not correctly analyzed and it is derived from a loan word. Since it is a loan word, it doesn't obey Kazakh language rules.

Table 5. Test Results of Morphological Disambiguator

		Correctly Disambiguated		
Files	Total Words	Before Rules Applied	After Rules Applied	Precision
1.txt	6462	4588	5621	0.870
2.txt	3124	2249	2749	0.880
3.txt	2836	1956	2412	0.851
4.txt	2532	1774	2177	0.860
Total	14954	10567	12959	0.867

For the morphological disambiguator, a training corpus with 30171 words is used and all words in this training corpus are manually tagged with their correct morphological parses. From this training corpus, our morphological disambiguator learned 512 disambiguation rules. The corpus used for the morphological analyzer is used a test corpus for our morphological disambiguator. This test corpus contains four files and 14,954 words in total. The results of disambiguated files are given in Table 5. 12,959 words of the test corpus with 14,954 words are correctly disambiguated. Without using the learned rules, 10,567 words are disambiguated just using most likely tags of words. Thus, 2,392 words are corrected by learned rules. The precision value for our morphological disambiguator is 0.87 percent. The accuracy can be raised by adding hand crafted rules to the disambiguation system.

8 Conclusion

Language is one of the main tools for communication. Thus, its investigation will provide better perspectives on all other aspects related with NLP. However, the formalization and computational analysis of Kazakh language morphology are not widely worked out. In other words, there is lack of tools for analysis of Kazakh language morphology from computational point of view. Moreover, grammar resources contain variances depending on scholars. For example, in some resources there are twelve tenses, whereas in others there are much less tenses of verbs. Naming of tenses can also vary from source to source. To summarize, building correctly working system of morphological analysis by combining all information is valuable for further researches on language.

In this paper, a detailed analysis of Kazakh language has been performed. Also, the formalization of rules over all morphotactics of Kazakh languages is worked out. By combining all gained information, a morphological processor is constructed. For the future work, enhancing of morphological analyzer should be performed by adding exception rules for widely used loan words. Also, performance of disambiguation system should be enhanced. In our system, it produces 87% accuracy and it should be enhanced up to 98% by adding some rules. Moreover, releasing the working system to users on the web and collecting feedbacks are intended. These feedbacks from users can help to improve the system capacity and lessen any possible errors.

Acknowledgements

This paper is a revised and extended version of our MORPHFSM-2014 conference paper whose title is "Rule Based Morphological Analyzer of Kazakh Language" [17].

REFERENCES

[1] G. Altenbek, X. Wang. Kazakh Segmentation System of Inflectional Affixes. Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP2010), Beijing, China, 183–190, 2010.

[2] K. Altintas, I. Cicekli. A Morphological Analyser for Crimean Tatar, Proceedings of the 10th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'2001), North Cyprus, 180-189, 2001.

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

[4] K.R. Beesley, L. Karttunen. Finite State Morphology, CSLI Publications, Stanford, CA, 2001.

[5] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, Computational Linguistics, 21(4), 543-566, 1995.

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

[7] D. Cutting, J. Kupiec, J. Pealersen, P. Sibun. A practical part-of-speech tagger, In Proceedings of the Third Conference on Applied Natural Language Processing, Trento, Italy, 1992.

[8] T. Daybelge, I. Cicekli. A Rule-Based Morphological Disambiguator for Turkish, Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), Borovets, Bulgaria. 145-149, 2007.

[9] K. Demirci. Kazakh Verbal Structures and Descriptive Verbs, Dunwoody Press, USA, 2006.

[10] D. Z. Hakkani-Tur, K. Oflazer,G. Tur. Statistical Morphological Disambiguation for Agglutinative Languages, Computers and the Humanities, 36(4), 381-410, 2002.

[11] M. Hulden. Foma: a finite-state compiler and library, Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session, Association for Computational Linguistics, 29-32, 2009.

[12] S. Isaeva, G. Nurkina. Sopostavitelnaya tipologiya kazakhskogo i russkogo yazykov, Ucheb. posobie, Sanat publishers, Almaty, Kazakhstan, 1996.

[13] O. Istek, I. Cicekli, A Link Grammar for an Agglutinative Language, Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), Borovets, Bulgaria, 285-290, 2007.

[14] M.B. Kairakbay, D.L. Zaurbekov, Finite State Approach to the Kazakh Nominal Paradigm, Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing (FSMNLP 2013), Scotland, 2013.

[15] M.A. Karaev. Qazaq tili, Almaty : Ana tili, 1993.

[16] L. Kartunen, R. M. Kaplan, A. Zaenen. Two-Level Morphology with Composition, Proceedings of the 14 th International Conference on Computational Linguistics (COLINGapos 1992).Nantes, France, 141-148, 1992.

[17] G. Kessikbayeva, I. Cicekli. Rule Based Morphological Analyzer of Kazakh Language, Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM (MORPHFSM 2014), Baltimore, USA, 46-54, 2014.

[18] M. Kutlu, I. Cicekli. A Hybrid Morphological Disambiguation System for Turkish, Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan, 2013.

[19] O. Makhambetov, A. Makazhanov, I. Sabyrgaliyev,Z. Yessenbayev.Data-Driven Morphological Analysis and Disambiguation for Kazakh.Computational Linguistics and Intelligent Text Processing - 16th International Conference, (CICLing 2015), Cairo, Egypt, 151-163, 2015.

[20] I.E. Mamanov. Qazaq til biliminin maseleleri, Aris publishers, Almaty, Kazakhstan, 2007.

[21] M. Mohri. On some applications of finite-state automata theory to natural language processing, Natural Language Engineering, 2(1), 61-80, 1996.

[22] M.K. Mussayev. The Kazakh Language, Vostochnaya literatura publishers, Moscow, Russia, 2008.

[23] K. Oflazer. Two-level Description of Turkish Morphology, Literary and Linguistic Computing, 9(2), 137-148, 1994.

[24] K. Oflazer, I. Kuruoz. Tagging and morphological disambiguation of Turkish text. In Proceedings of the 4th Applied Natural Language Processing Conference, 144-149, 1994.

[25] K. Oflazer, G. Tür. Morphological Disambiguation by Voting Constraints. In Proceedings of ACL/EACL, The 35th Annual Meeting of the Association for Computational Linguistics, Madrid, Spain, 1997.

[26] M. Orhun, C. Tantug, E. Adali. Rule Based Analysis of the Uyghur Nouns, International Journal of Asian Lang. Proc.,19(1), 33-44, 2009.

[27] Qazinform. www.inform.kz National news agency, Kazakhstan, 2010.

[28] H. Sak, T. Güngör, M. Saraçlar. Morphological disambiguation of Turkish text with perceptron algorithm. In Proceedings of CICLing, 107-118, 2007.

[29] C. Tantug, E. Adali, K. Oflazer. Computer Analysis of the Turkmen Language Morphology, 5th International Conference on NLP(FinTAL 2006), Turku, Finland, 186-193, 2006.

[30] P. Tapanainen, A. Voutilainen. (1994). Tagging Accurately - Don't guess if you know. In Proceedings of ANLP94, 47-52, 1994.

[31] Z.Q. Tuymebayev. Qazaq Tili: Grammatikaliq aniqtagish, Almaty, Kazakhstan, 1996.

[32] T. Valyaeva. Kazakhskii yazyk, http://kaz-tili.kz/. 2014.

[33] J. Washington, I. Salimzyanov, F. M. Tyers, Finite-state morphological transducers for three Kypchak languages, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, 3378-3385, 2014.