Rule Based Analysis of the Uyghur Nouns

Murat Orhun*, A.Cüneyd Tantuğ**, Eşref Adalı**
* Computer Science Dept.Isntanbul Bilgi University.
Kurtuluş Deresi Cad No:47 Dolapdere 34440 Beyoğlu
Istanbul, Turkey
muratmehmet@cs.bilgi.edu.tr
** Computer Engineering Dept. Istanbul Technical University.
80626 Maslak, Instanbul Turkey
{cuneyd,adalı}@itu.edu.tr

Abstract:

This paper describes the implementation of a rule-based analyzer for Uyghur (spoken in Sin Kiang, China) Nouns. We hope this paper will give some contribution for advanced studies to the Uyghur Language in Machine Translation and Natural Language Processing. Like all Turkic languages, the Uyghur Language is an agglutinative language that has productive inflectional and derivational suffixes. In this work, we implemented a finite state two-level morphological analyzer for the Uyghur Nouns by using Xerox Finite State Tools.

Keywords:

Uyghur nouns, machine translation, Uyghur morphology, Uyghur lexicon, Uyghur grammar.

1 Introduction

This paper describes the implementation of a rule-based analyzer for the Uyghur nouns. We use two-level morphological analysis, which is widely used for analyzing morphology (Koskenniemi, 1985). Uyghur language belongs to the Qarluq group of the Turkic language family (wikipedia, 2008 and Kaşgarlı, 1992). The morphology of the Uyghur language is similar to other Turkic languages. The agglutinative structure of the language exhibits very productive inflectional and derivational morphology. Hence, morphological analysis is a key component in most of the NLP related applications on languages with rich morphology. There are a number of researches in the literature which have focused on two-level analysis of various languages like Japanese (Alam, 1983), Finnish (Koskenniem, 1985) and Romanian (Khan, 1983). However, there exists limited NLP research on Turkic languages except than Turkish (Altintaş, 2001; Tantuğ, 2007 and Oflazer , 1994). The most recent research about Turkic language is about Turkmen language (Tantuğ, 2006). Even though Turkic languages are similar, there exist great divergences like different tenses, different subject-verb agreements and etc. In this work, we implemented a finite state two-level morphological analyzer for the Uyghur nouns by using Xerox Finite State Tools (Karttunen, 1997). In the second section we give some information about the Uyghur Language and its orthography, and then the following section describes a brief overview of two-level morphology. In the fourth section, the two-level rules developed for Uyghur are explained

and exemplified in detail. In the fifth section we describe the finite state machines for morphotactics. In the sixth section we give some examples about the analyses of the Uyghur nouns and evaluate our Finite state machine. The last section concludes the results of the work and gives some suggestion for further work.

2 The Uyghur Language

The Uyghur Language is a Turkic language which belongs to the Ural-Altaic language family and has approximately 10 million native speakers. In history, Uyghur people invented and used different alphabets (wikipedia, 2008 and Kaşgarlı, 1992) For example, Orhun - Yinsey Alphabets (Kök Türk Scripts), Old Uyghur Alphabets etc. "Old Uyghur" is a name sometimes applied to the Sogdian alphabet, originally used for Sogdian (an Iranian language). From the 9th to the 13th century, it was used for writing the Old Uyghur language, especially in Buddhist texts. Forms of it survived till the 18th century for other languages used in Sin Kiang. It is ultimately derived from the Syriac alphabet, but differs from it by being written from top to bottom instead of left to right. The Man-chu and Mongolian alphabets are descended from it (wikipedia, 2008 and Kaşgarlı, 1992). The Uyghur language traditionally used a modified Perso-Arabic alphabet, known as Chagatai script or Kona Yeziq (old script), since the 10th century. The Chinese government introduced a Roman script (Yengi Yeziq, “new script”) closely resembling the Soviet Uniform Turkic Alphabet in 1969. A further modification of the Arabic script, with additional diacritics to distinguish Uyghur vowels, was introduced in 1983: this is known as Uyghur Ereb Yéziqi (Uyghur Arabic script) and is still widely used. Cyrillic script has been used and is in parts still being used to write Uyghur in areas previously dominated by Russians, and another Roman script, based on Turkish orthography, is used in Turkey and on the internet (wikipedia, 2008 and Kaşgarlı, 1992). There is a comparison of Uyghur alphabets are given at (wikipedia, 2008 and Kaşgarlı, 1992). Currently, Uyghurs in Sin Kiang use the Arabic alphabets. The latest researches about Uyghur alphabets are could be found in (Duval, 2006 and Saleh, 2006). The Uyghur Computer Science Association is formed by volunteer group in order to contribute the researches on Uyghur language in computer technology (UKIJ, 2005), by trying to develop Uyghur alphabets. This association has implemented some software which translates different Uyghur alphabets each other. The current Uyghur orthography is composed of 32 Latin letters. This letters standardized by the UKIJ (2005). The detailed information about the characters are described in (Duval, 2006 and Saleh, 2006). These letters are:
a e b p t j ch x d r z j s sh gh f q k g ng l m n h o u ö u ü w ė i y. There are 8 vowels (a e ė i o ö u ü) and 24 consonants (b p t j ch x d r z j s sh gh f q k g ng l m n h w y).
The Uyghur vowels are very complicated. For example, the two vowels “i, ė ” sometimes perform as front vowel, or sometimes performs as back vowel. These two vowels are categorized as middle vowels. Consonants are categorized as voiced and unvoiced consonants. But all vowels could be recognized as resonant or voiced consonants (Hamit, 2003).

3 Two Level Morphology

Two-level morphology is a widely used technique in morphological analysis. Especially it

is very useful analyzing agglutinative languages. There are two levels called lexical and surface levels. In the surface level, a word is represented in its original orthographic form. In the lexical level, a word is represented by denoting all of the functional components of the word. The phonetic modifications and restrictions are represented using four different rule types (Sproat, 1992).

`1. a:b => LC_RC`	A is realized as b only in the context LC (left con-text) and RC (Right Context), but not necessarily.
`2. a:b <= LC_RC`	A is **always** realized as b in the context LC and RC.
`3. a:b <=> LC_RC`	A is **always** realized as b in the context LC and RC and nowhere else.
`4. a:b /<= LC_RC`	A is **never** realized as b in the context LC and RC.

User defined rules of these types are used to generate a finite state acceptor. This machine accepts a legal lexical surface matching whereas it rejects an illegal matching. Detailed explanation and examples about these four rule types can be found in (Osmanof, 1997 and Xiu, 1998).

4 Two Level Rules

We have defined an alphabet for the two-level description of the Uyghur language. This alphabet includes the standard Uyghur letters and some additional symbols which are used in the intermediate level and have no usage in orthography. We have represented the non-ASCII Uyghur letters by their uppercase counterparts such as, ö→O, ü→U, ê→E. In UKIJ, there are some letters such as ng, gh, sh, ch, j(zh) which are represented by two Latin letters whereas each of these are single Arabic letters. We have represented them uppercase characters such as ch→c, sh→S, ng→N, gh→G. j→Z. In UKIJ, they use the "j" letter to represent j and zh at the same time. But, we use "Z" to represent "zh", which is the counterpart of "j" in UKIJ.

To define some two- level rules for Uyghur nouns, we have classified Uyghur letters into some groups according to Uyghur grammar sources as instructed in (Kaşgarlı, 1992; Hamit, 2003; Osmanof, 1997 and Xiu, 1998).

```
Consonants:
CONS = b p t j c x d r z Z s S G f q k g N l m n h w y
Voiced consonants:
CONSVO = b j d r z Z G g N l m n h w y
Unvoiced consonants:
CONSUVO = p t c x s S f q k
Consonants which makes the Middle vowel to Front vowel:
CONSFV = k g
Consonants which makes the Middle vowel to Back vowel:
b p t j c x d r z Z s S G f q N l m n h w y
Vowels:
VOWEL = a e E i o O u U
Back Vowels:
BACKV = a o u
Front Vowels:
```

```
FRONTV = e O U
Middle Vowels:
MIDV = E i
Labialized Vowels:
LABV = o O u U
Non-labialized Vowels:
NLABV = a e E i
Back rounded vowels :
Lou = o u ;
Front rounded vowels:
LOU = O U ;
Unrounded vowels:
LeiE = e i E a
```

Apart from the ordinary letters, we have defined some consonants and vowels for lexical level. These letters used only lexical level and not appear on the surface:

```
Lexical Vowels:
A = a, e
H = i, u, U
Lexical Consonants:
D = d, t
Y = G, q, g, k
```

Our analyzer involves a number of rules to generate the realizations of lexical level representations. An interesting morphophonetic rule which is observed in Uyghur as well as most of the Turkic languages is the vowel harmony rule. As an example, the plural suffix which is represented by +lAr in lexical level have two different surface level realizations +lar and +ler determined by the previous vowels. If the preceding vowel (may be a the stem or in a preceding suffix) is a front vowel, the plural suffix +lAr is is resolved as “+lar”; if the preceding vowel belong to the back vowels set, it is resolved as ”+ler”. If the preceding vowel is a middle vowel, then the situation is a bit more complicated where existence of some constants should be taken into account to decide the proper surface realization.

1.
```
A:a => [:BACKV][:CONS]*(%+:0)[:CONS|CONS:]* _  ;
[[:CONSBV][:MIDV]]|[[:MIDV][:CONSBV]]+(%+:0)(K:0)[:CONSBV
|CONSBV:]* _ ;
```

2.
```
A:e =>[:FRONTV][:CONS]*(%+:0)[:CONS |CONS:]* _ ;
[[:CONSFV]+[:CONS]*[:MIDV]]|[[:CONS][:MIDV][:CONSFV]]+(%+:0)
[:CONS|CONS:]* _ ;
[:CONS]*[:MIDV][:CONS]*(M:0)(%+:0)[:CONS|CONS:]* _ ;
```

These two rules are for vowel harmony rules.

```
Lexical: kigiz+lAr
Surface: kigiz0ler
```

In this example, last syllable of the noun “kigiz” is “giz”, and it includes the middle vowel “i”. The plural suffix is resolved as “ler” according to rules.

```
Lexical: bEnit+lAr
Surface: bEnit0lar
```

In this example, last syllable of the noun "bEnit" is "nit", and it includes the Middle vowel "i". This time, the plural suffix is resolved as "lar". The ownership – dependent category of the noun is the grammatical category which indicates that the object expressed by the noun is dependent on (or belongs to) a certain (other) object. In Uyghur, this category is expressed by the governorship- depen-dent forms that made by adding the noun's ownership-dependent suffixes. The ownership-dependent suffixes of nouns in Uyghur are:
+m, +im, +um, +Um, +miz, +imiz, +N, +iN, +uN, +UN, +Nlar, +iNlar, +uNlar, +UNlar, +Niz, +iNiz, +liri, +i, +si

The rules 3-7 in below define ownership-dependent for Uyghur nouns.

```
3.H:I<=>[:CONS]*[:LeiE][:CONS](M:0)[:CONS|Y:]*(%+:0) _ ;

4. H:u <=> [:CONS]*[:Lou][:CONS|CONS:]+(%+:0) _;

5. H:U <=> [[:CONS]*[:VOWEL]*]*[:LOU][:CONS]+(%+:0) _  ;
```

In these three rules, H forced to harmonize with other vowels.

```
6. H:u /<= :CONS]*[:VOWEL][:CONS]* (%+:0) n _ N
[:CONS]*[:VOWEL][:CONS]* (%+:0)n _ ;

7. H:U /<= :CONS]*[:VOWEL][:CONS]* (%+:0) n _ N
[:CONS]*[:VOWEL][:CONS]* (%+:0)n _;
```

These two rules prevent the appearance of "H" as "u" or "U" on the surface level. Otherwise, there will be a conflict with the rule 3.
For example:

```
Lexical: qol+nHN
Surface: qol0niN (according to rule 3). (of the hand)

Lexical: qol+nHN
Surface: qol0nuN (incorrect, and prevented by rule 6)

Lexical: qol+Hm
Surface: qol0um (according to rule 4) (my hand)
```

```
8. H:0 => [:CONS]*[:VOWEL](%+:0)_ ;
```

H vowel is deleted if the last letter of the stem is being affixed to is a vowel. For example:

```
Lexical: naxSa+Hm
Surface: naxSa00m (my song)
```

There are six cases for Uyghur nouns. The case category of the nouns indicates the syntactical relationships the noun and other words. The seven cases and their suffixes are:

```
Nominative case: 0 (no suffix)
Possessive case: niN
Dative case: Ga, qa, ge, ke
Accusative case: ni
Locative case: da, ta, de, te
Ablative case: din, tin
Similitude case: dek, tek

9. D:t =>[:CONS]*[:VOWEL][CONSUVO|Y:][:VOWEL]+(%+:0)_ ;

10.D:d =>[:CONS]*[:VOWEL][CONS]*[CONSVOaz][:VOWEL]*(%+:0)_;
```

These two rules are harmonization of the D consonants with other consonants.
For example:

```
Lexical: meydan+DA
Surface: meydan0da (at the square)

Lexical: ders+DA
Surface: ders0te (in the course)

11. Y:q =>[:CONS]*[o| u | a|A:][:CONSUVO|CONSUVO:]+(%+:0) _  ;
          G(%+:0) _ [:VOWEL][:CONS]* [.#.]

12. Y:k => [:CONS]*[: [e | i | E | U |O ]][CONSUVO]+(%+:0) _ ;

13. Y:G => [[:CONS]*[o | u | a|H:i]]+[:CONSVOaz]* (%+:0) _  ;
           [:CONS]* i d i (%+:0) _ ;

14. Y:g => [[:CONS]*|[.#.]] [e | i | E | U |O ][CONSVOaz](M:0)
           (%+:0) _;
```

These rules are for the harmonization of Y consonants with the other consonants and vowels. The harmonization of the Y consonants is tricky because, its harmonization is defined by the consonants and vowels at the same time.

```
Lexical:  medan+YA
Surface: meydan0Ga (to square)

Lexical:  kitab+YA
Surface: kitab0qa (to a book)

Lexical:  Oy+YA
Surface:  Oy0ge (to home)

Lexical:  ders+YA
Surface:  ders0ke (to class)
```

```
15. a:E <=> .#. [:CONS]* _ [CONS] (%+:0) [[H:i] m |[H:i] N |
[H:i] | [H:i] m [H:i] z |[H:i] N [H:i] z ] ;
```

This rule describes that the vowel “a” in a monosyllable stem should be transformed to vowel “E” when this stem is followed by a suffix whose first vowel is “i”. For ex-ample:

```
Lexical: can+Hm
Surface: cEn0im (my dear)
```

```
16. e:E <=> .#. [:CONS]* _ [CONS] (%+:0) [[H:i] m |[H:i] N |
[H:i] | [H:i] m [H:i] z |[H:i] N [H:i] z ] ;
```

This rule describes that the vowel “e” in a monosyllable stem should be transformed to vowel “E” when this stem is followed by a suffix whose first vowel is “i”.
For ex-ample:

```
Lexical: bel+Hm
Surface: bEl0im (my back)
```

```
17.a:i<=>[CONS]*[VaeE][CONS][[CONS]*[VaeE][CONS]]*[CONS]*_
[CONS]* (%+:0) [[H:i] m |[H:i] N | [H:i] | [H:i] m [H:i]z
|[H:i] N [H:i] z ];
[CONS]*[VOWEL][CONS] _ s [H:i];
```

```
18.e:i<=>[CONS]*[VaeE][CONS][[CONS]*[VaeE][CONS]]*[CONS]*_
[CONS]* (%+:0) [[H:i] m |[H:i] N | [H:i] | [H:i] m [H:i] z
|[H:i] N [H:i] z ];
```

These two rules describe that the vowels “e” and “a” in a stem having two or more syllables are realized as the vowel “i” when this stem is followed by a suffix whose first vowel is “i”. For example:

```
Lexical: taGlar+Hm
Surface: taGlir0im (my mountains)
Lexical: qelemler+Hm
Surface: qelemlir0im (my pencils)
```

```
19. [u:0 | i:0 ] <=> [CONS]*[$:0] _ [CONS][CONS]*[VOWEL]*
(%+:0)H;
```

This rule describes the “u” and “v” vowels deleted in the some nouns when some suffixes are added. Since this deletion is not a general rule and applies only for some of the nouns, the vowels which should be deleted in such context are denoted by a “$” sign in the lexicon. For example:

```
Lexical: bur$un+Hm
Surface: bur00n0um (my nose)
```

```
Lexical: EG$iz+HN
Surface: EG00z0iNm (your mouse)
```

5 Morphotactics

The study and modeling of legal word formation is called morphotactic (Kenneth, 2003). In agglutinative languages the morphemes are affixed to a root word like"beads on a string" (Sproat, 1992). In our implementation, morphotactics are done by finite-state-machines. The finite state machine for nominal morphotactics is depicted in Figure 1.

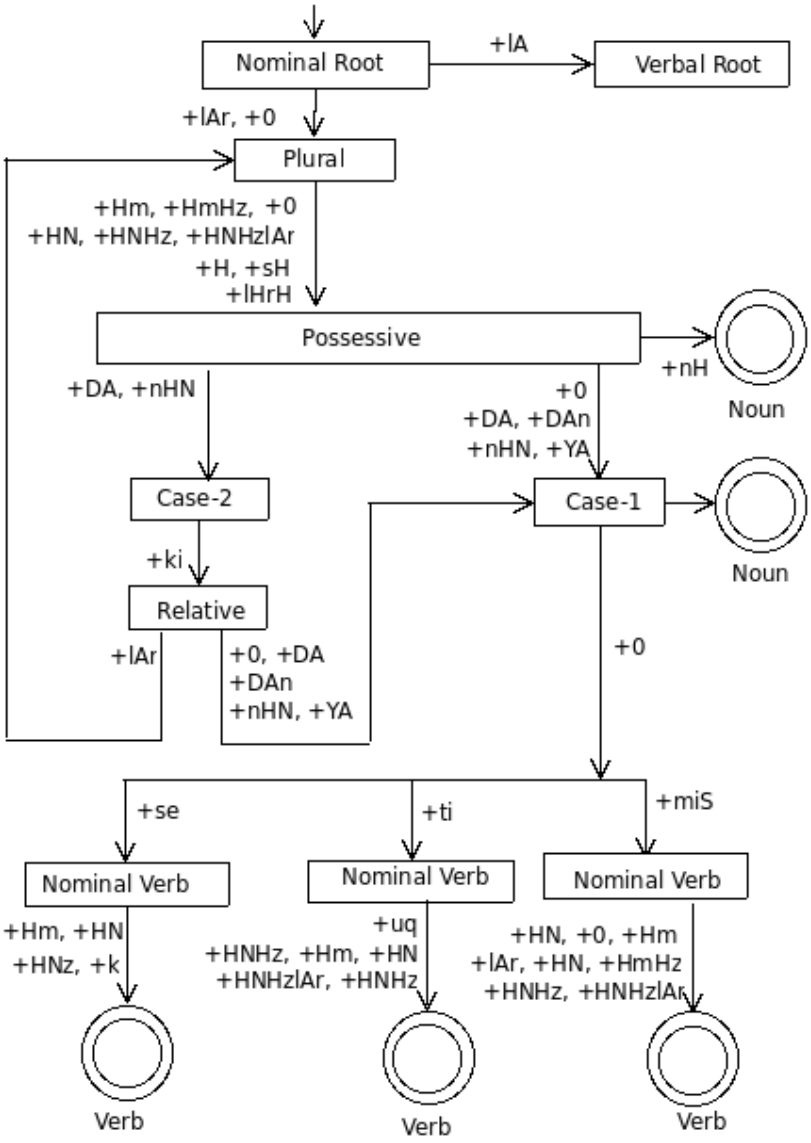

Figure 1. Finite State Machine for Nominal Morphotactics

In this figure, the boxes show the states, the arrows show the next state that can be reached when a suffix matching one of the labels are found. The circles are the final states which indicate legal word formations. The 0 transitions indicate that the transition can be done without adding any suffix. The XFST environment has a module called LEXC to build the finite-state-machines as morphotactic rules. A small section of the LEXC lexicons are given below:

```
LEXICON Noun

kitap           NounPOS ;
at              NounPOS;
su              NounPOS;

LEXICON         NounPOS
+Noun:0Suffixes;

LEXICON         Suffixes
+Plural:lAr      Accusative;
+Plural:lAr     Dative;

LEXICON         Accusative
+Acc:nH          Final;

LEXICON         Dative
+Dat:YA          Final;

LEXICON         Final
        #;
```

6 Evaluation

We have implemented the morphologic analyzer for Uyghur nouns with the Xerox Finite State Tools (Karttunen, 1997). We have tested the analyzer with different nouns and the affixes on the Fig 1. For example if we input the noun “kitap” (book), we get the following results according to different suffixes.

```
Input: kitap (book)
Ouput: kitap+Noun+A3sg+Pnon+Nom
```

“A3sg” stands for number person agreement and it means third person single. All of the single nouns are marked as third person single. “Pnon” stands for none possessive agreement. If the noun is possessed by the first person, then it is marked as “P1sg”.
“Nom” stands for nominative.

```
Input: kitaplAr (books)
Output: kitap+Noun+A3pl+Pnon+Nom
```

“A3pl” stands for the third person plural

```
Input: kitaplArHN (your books)
Output: kitap+Noun+A3pl+P2sg+Nom
```

“P2sg” stands for the second person single

```
Input: kitaplArHmHzDA (in our books)
Output: kitap+Noun+A3pl+P1pl+Loc
```

“P1pl” stands for the first person plural. “Loc” stands for locative suffix.

```
Input: kitaplArHmHzDAkinHN (of the our books)
Output: kitap+Noun+A3pl+P1pl+Loc^DB+Adj+Rel+Gen
```

“DB” mean derivation. Here Adjective is derived from a noun and marked as “Adj”.
“Gen” stands for genitive suffix.

```
Input: kitapnHDA
Output: NONE
```

“None” stands for no resolution for the input words.
In this example, the noun “kitap” is followed by the “nH” suffix and “DA” suffixes. In Uyghur Language the “nH” (accusative) suffixes never followed any other suffixes (Kaşgarlı, 1992; Hamit, 2003; Osmanof, 1997 and Xiu, 1998).
So the word “kitapnHDA” is an incorrect word.
As a result, we get the satisfactory result that we expecting from our analyzer.

7 Conclusion

This work gives some implementation details of rule-based morphological analysis of the Uyghur nouns. We have used two-level morphology approach with Xerox finite state machines. As far as we know, this is the first effort on computational analysis of Uyghur morphology. Having a morphological analyzer is a very critical issue especially for NLP related tasks on agglutinative languages. This work is limited in the way that it covers only the nouns in Uyghur language. Al-so the noun root lexicon should be extended to be able to build a general purpose wide-coverage morphological analyzer. We are still working on both adding new root words and building the finite state machines for morphological analyzing of the other POS classes, especially verbs which is much more complex and troublesome.

8 References

Alam, Y.S : *A Two-Level Morphological Analysis of the Japanese*. Texas Linguistics Forum, 22:229-252 (1983).

Altintas, K. Cicekli, İ.: *A Morphological Analyzer for Crimean Tatar*. Proceedings of the 10th Turkish Symposium on the Artificial Intelligence and Neural Networks (TAINN2001), North Cyprus, pp: 180-189 (2001).

Duval J.R., Janbaz, W, A. : *An Introduction to Latin-Script Uyghur*. Middle East & Central Asia Politics, Economics, and Society Conference. Sept 7-9, University of Utah, Salt Lake City, USA (2006).

Hamit T. : *Modern Uyghur Grammar (Morphology)*. Yildiz Teknik Üniversitesi, Fen-Ed Fak. T.D.E Bölümü. Istanbul (2003).

Karttunen, L., Gaal, T., Kempe, A. : *Xerox Finite State Tool. Technical Report*, Xerox Research Centre, Europe (1997).

Kaşgarlı S.M.: *Modern Ugur Türkçesi Grammeri*, İstanbul (1992).

Koskenniemi, K.: *Two-Level Morphology : A General Computational Model for Word Form Recognition and Production*. Publication No:11 , Department of General Linguistics, University of Helsinki (1985).

Koskenniemi, K. : *An Application of the Two-Level Model to Finnish*. In Fred Karlsson, editor, Computational Morphosyntax, a report on research 1981-1984. University of Helsinki Department of General Linguistics (1985).

Kenneth, B.R., Karttunen, L. : *Finite State Morphology*, CSLI Publications (2003).

Oflazer, K. : *Two-Level Description of Turkish Morphology*. Literary and Linguistic Computing, Vol. 9, No:2 (1994).

Oflazer, K. and Kuruöz, I. : *Tagging and Morphological Disambiguation of Turkish Text*, Proceedings of the 4th ACL Conference on Applied Natural Language Processing, Stuttgart, Germany, (1994).

Osmanof, M. : *Hazirqi Zaman Uyghur Edebiy Tilining İmla ve Teleppuz Lughiti*. Shin Jiang Xeliq Neshiryatı. Ocak. (In Uyghur) (1997).

R. Khan, *A Two-Level Morphological Analysis of Rumanian*. In Texas Linguistic Forum, Texas, USA, pp. 253-270, (1983).

Saleh, I., Janbaz, W.A. : *Web Development Considerations for Unicode-based Text Processing in Uyghur Language*. The 30th Internationalization and Unicode Conference, Washington, DC USA, (2006).

Sproat, R. : *Morphology and Computation*. MIT Pres., (1992).

Tantug, A.C. : *A Hybrid Model For Machine Translation Between Agglutinative and Related Languages*. PhD Thesis, Istanbul Technical University, (2007).

Tantuğ A.C., Adalı E., Oflazer K. : *Computer Analysis of The Turkmen Language Morphology*. Proceedings of the 5th International Conference on Natural Language Processing, FinTAL 2006, Turku, Finland, (2006) .

UKIJ. *Uyghur Komputer Ilmi Jemiyeti*. http://www.ukij.org/.

http://en.wikipedia.org/wiki/Uyghur_language, (accessed on 04/2008).

Xiu Y.S., Ju, G.X., : *Uyghur Dili Grammatkası*. People's Republic of China National Central University Press (In Chinese) (1998).