TAGSET FOR NOOJ TURKISH MODULE

UMUT DEMİRHAN AND MUSTAFA AKSAN

Abstract

This study begins with an introduction about the complex morphotactics of Turkish and then focuses on currently available morphological analysers and their tagsets for Turkish. Finally, it presents the proposed tagset for NooJ Turkish Module. The performance and application of the current Tagset are also demonstrated through sample annotations.

Introduction

Tagging is an essential component in determining the linguistic structures in a corpus. It is almost impossible to conduct any linguistically significant study on a large text corpus without providing relevant information for the words that make up the texts. Annotating the words or assigning their essential grammatical information makes relatively higher level analyses possible including syntactic and semantic parsing.

In all languages of the world however, words are notoriously ambiguous in the sense that they belong to different word classes simultaneously. Thus, in tagging of the texts, what is expected from the tagger is not only to assign the proper part of speech tag to each and every lexical item in the text but also to resolve the potential ambiguities.

A note on Turkish

Turkish is an agglutinative language in which a set of morphemes may appear on the same root, each with a specific position and a function in the resulting structure. Most of the grammatical categories that are expressed by free morphemes in other languages are expressed by bound morphemes in Turkish. It is argued that distinguishing nominals from verbals is relatively easy and the phonological rules of harmony further help determine individual morphemes and morpheme boundaries clearly. Such properties further eliminate irregularity in lexical forms and morphological

rule applications. Homographic root forms are virtually non-existent, further ruling out potential part of speech ambiguities.

In any process of assigning a part of speech label for a word in Turkish, it should be noted that the boundaries between noun, adjective and adverb are "blurred" (Göksel and Kerslake 2005). In other words, there are lexical items that serve the typical function of any of these parts of speech:

- *aptal* 'stupid' (ADJ)
- *aptal-lar* 'the stupid/stupids' (Npl).

Strict morphological constraints on morpheme ordering and the phonological constraints that determine the shape and the boundaries of morphemes affixed to root most often resolve parts of speech ambiguities. However, the homographic morphemes and phonological changes that morphemes must undergo, due to various rules of harmony that apply in all affixation processes, create a different type of ambiguity in which a lexical item may be assigned different morphological analyses. This is mainly due to different interpretation of the morpheme in question or the boundary which in turn affects the part of speech interpretation of the root form.

Derived and nonderived homographic forms

- *kale-m* 'castle'-1st person possessive, 'my castle'
- *kalem* (non derived root form)

Homophonous affixes

- *bıçak-la* 'knife-with' 'with a knife'
- *bıçak-la* 'knife-verbalizer' "lit. to knife" "to wound or kill by using knife, stabbing"

In the rest of this paper we will present currently used morphological analysers and their tagset, proposed tagset for NooJ Turkish Module and our initial results.

Currently available morphological analysers and tagsets

Zemberek (Akın & Akın 2007) is one of the most popular morphological analysers of Turkish. Zemberek is an "*open source, platform independent, and general purpose Natural Language Processing library and toolset designed for Turkic languages, especially for Turkish*" (Akın & Akın

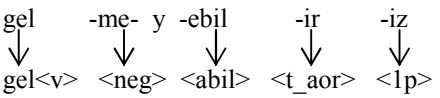
2007). Zemberek is officially used as a spellchecker in Open Office Turkish version and Turkish National Linux Distribution, Pardus. The application provides basic NLP operations such as spell checking, morphological parsing, and word suggestion (Akın & Akın 2007).

Zemberek also has its own tagset for tagging both stems and affixes; however, it does not allow researchers to search for the affixes and word forms with their tags on its web-interface, and it should be noted that Zemberek uses just letters for tags not the numerals. Sample tagging output of Zemberek for the word *gelmeyebiliriz* 'we may not come' is below:

- Root: *gel* (to come) Type: Verb
  V_Negation_mE_Auxiliary_ebil_Aorist_(i)r _Person_(i)z

TRmorph (Çöltekin 2010) is another popular morphological analyser of Turkish which is relatively complete for Turkish. It is implemented by using Stuttgart Finite State Tools (SFST) (Schmid 2005), and the application uses a lexicon, based on the word list from Zemberek. Although highly modified, the morphological analyser is distributed under GPL. In order to use the analyser, a researcher should have SFST. The analyser can also be compiled and used with Helsinki Finite State Tools (HFST) (Linden et al. 2009).

Çöltekin (2010) argues that the best feature of the application is the availability of two-level description of Turkish for researchers to use and modify it freely for different applications. It is also argued in his study that TRmorph is the first freely available morphological analyser for Turkish. TRmorph uses both numerals and letters for tags. TRmorph has its own tagset, however, it has only a few derivational affixes, and uses a very limited lexicon. A sample tagging result of the application for the word *gelmeyebiliriz* 'we may not come' is stated below:



Tagset for NooJ Turkish Module

Apart from the studies mentioned above, the latest release of *NooJ Turkish Module* has its own tagset too. The tagset is generated by using the data of a subcorpus extracted from Turkish National Corpus (TNC) Project. The subcorpus for the purpose of generating a tagset contains five

million tokens derived from written and spoken texts. The distribution of the subcorpus is stated below:

- 1.000.000 words from fictional texts
- 1.000.000 words from newspapers
- 1.000.000 words from journals
- 1.000.000 words from informative texts
- 500.000 words from unpublished written texts
- 500.000 words from spoken texts

Before the annotation process as in Silberztein (2003), first of all, the graphs of the module are created. Then dictionaries are stabilized with Tokenization, Lemmatization, Stating Phonetic Alternations, and Lexical Features. After achieving these, three different high level tagsets for NooJ Turkish Module are generated:

- High Level Tagset for Nominals which contains 11 tags
- High Level Tagset for Verbs which contains 1 tag
- High Level Tagset for the Others which contains 5 tags.

Table 1 located below contains the High Level Tagset for Nominals, Table 2 lists High Level Tagset for Verbs, and lastly Table 3 shows the High Level Tagset for the Other Part-of-Speech Tags for NooJ Turkish Module.

TAG	Part-of-Speech
N	Noun
A	Adjective
PN	Pronoun
NP	Proper Name
AB	Abbreviation
AV	Adverb
PP	Postposition
DET	Determiner
NU	Number
ON	Onomatopoeia
CL	Clitics

Table 1: High Level Tagset for Nominals

TAG	Part-of-Speech
V	Verb

Table 2: High Level Tagset for Verbs

TAG	Part-of-Speech
CJ	Conjunction
IJ	Interjection
Q	Question
FR	Foreign Words
ER	Spelling Error

Table 3: High Level Tagset for the Others

On the other hand, the low level tagset of NooJ Turkish Module does not contain the affixes included in derivational paradigm. In low level tagset, we have two subcategories called Nominal Affix Tagset and Verbal Affix Tagset. In nominal paradigm, we have 50 different tags, and in verbal paradigm we have 48 different tags for the Tagset of NooJ Turkish Module. Table 4 below shows the affixes in the first column, the function of the affix is stated in the second column, and the third column shows the tag used for the affix for Inflectional-Nominal Paradigm. In addition, Table 5 lists the affixes, functions, and the tags of Inflectional-Verbal Paradigm of the Tagset for NooJ Turkish Module.

	AFFIX	FUNCTION	TAG
1	lAr	number/person	pl
2	I	buffer phoneme	bfi
3	n	buffer phoneme	bfn
4	(y)	buffer phoneme	bfy
5	(s)	buffer phoneme	bfs
6	(ş)	buffer phoneme	bfsh
7	NOMINATIVE	case	nom
8	I	case	acc
9	In[GEN]	case	gen
10	A[DAT]	case	dat
11	DA[LOC]	case	loc
12	DAn[ABL]	case	abl
13	ile	case	ins
14	Im[1Psn]	person_copula	c1s

	AFFIX	FUNCTION	TAG
15	Iz[1Ppl]	person_copula	c1p
16	sIn[2Psn]	person_copula	c2s
17	sInIz[2Ppl]	person_copula	c2p
18		person_copula	c3s
19	lAr[3Ppl]	person_copula	c3p
20	m[Poss]	possessive	p1s
21	mIz[Poss]	possessive	p1p
22	n	possessive	p2s
23	nIz[Poss]	possessive	p2p
24	I	possessive	p3s
25	lArI	possessive	p3p
26	i	verb	Vi
27	DIr	copula	cop
28	DI[Past]	copula	past
29	mIş[Perf]	copula	perf
30	m[1Psn]	person	1s
31	n[2Psn]	person	2s
32	k[1Ppl]	person	1p
33	nIz[2Ppl]	person	2p
34	[3Psn]	person	3s
35	lAr[3Ppl]	person	3p
36	sInIz[2Ppl]	person	2p
37	sIn[2Psn]	person	2s
38	Iz[1Ppl]	person	1p
39	Im[1Psn]	person	1s
40	mAk_NN	nominal	nz1
41	AcAk_NN	nominal	pc1
42	mA_NN	nominal	nz2
43	DIk_NN	nominal	pc2
44	An_AJ	adjectival	pc3
45	ki_AJ	adjectival	kiA
46	ki_PN	pronominal	kiP
47	cA_AV	adverbial	AV13
48	cAsInA_AV	adverbial	AV12
49	ken_AV	adverbial	AV10
50	sA_AV	adverbial	AV11

Table 4: Nominal Affix Tagset

	AFFIX	FUNCTION	TAG
1	(y)	buffer phoneme	bfy
2	(I)	buffer phoneme	bfi
3	A	buffer phoneme	bfa
4	yor	imperfective	iprf
5	bil	auxiliary verb	Va1
6	ver	auxiliary verb	Va2
7	dur	auxiliary verb	Va3
8	gel	auxiliary verb	Va4
9	gör	auxiliary verb	Va5
10	yaz	auxiliary verb	Va6
11	kal	auxiliary verb	Va7
12	koy	auxiliary verb	Va8
13	AyIm[IMP]	imperative	imp1
14	sIn[IMP]	imperative	imp2
15	AlIm[IMP]	imperative	imp3
16	In(Iz)[IMP]	imperative	imp4
17	sInlAr[IMP]	imperative	imp5
18	mA	negative	neg
19	ik[1Ppl]	person	1p
20	k[1Ppl]	person	1p
21	(I)z[1Ppl]	person	1p
22	(I)m[1Psn]	person	1s
23	nIz[2Ppl]	person	2p
24	sInIz[2Ppl]	person	2p
25	sIn[2Psn]	person	2s
26	n[2Psn]	person	2s
27	lAr[3Ppl]	person	3p
28	r[Aor]	aorist	aor
29	z[Aor]	aorist	aor
30	mAktA[Cont]	imperfective	cont
31	AcAk[Futr]	future	futr
32	mAlI[Necc]	necessity	necc
33	DI[Pas]	past / perfective	past
34	mIş[Per]	referential/perfective	perf
35	i	verb	Vi
36	DIr(P)	copula	cop
37	AlI_AV	adverbial	AV01
38	ArAk_AV	adverbial	AV02
39	ArAktAn_AV	Adverbial	AV03
40	AsIyA_AV	adverbial	AV04

	AFFIX	FUNCTION	TAG
41	DIkçA_AV	adverbial	AV05
42	IncA_AV	adverbial	AV06
43	Ip_AV	adverbial	AV07
44	mAdAn_AV	adverbial	AV08
45	mAksIzIn_AV	adverbial	AV09
46	ken_AV	adverbial	AV10
47	sA_AV	adverbial	AV11
48	cAsInA_AV	adverbial	AV12

Table 5: Verbal Affix Tagset

The implementation and performance of the tagset for NooJ Turkish Module is tested and evaluated by concordances and morphological annotation results. A sample annotation result for the word *'gelmeyebiliriz'* 'we may not come' is stated in Figure 1.

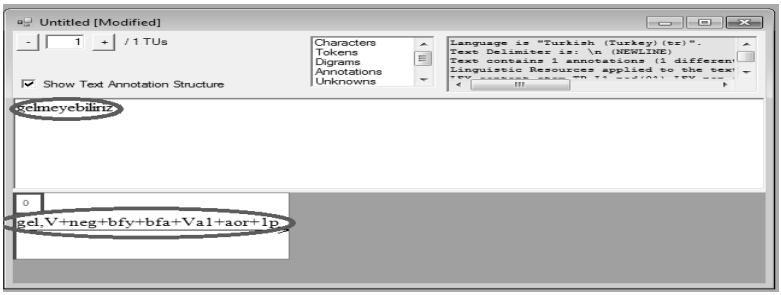


Figure 1: NooJ Turkish Module annotation result

Table 6 shows the performance of the tagset for NooJ_TR module. Number of types and unknown words are classified, and the tagset for NooJ_TR Module works successfully for 5 million words subcorpus used.

Consequently, it is also important to mention that there are not yet any other applications which have the capability of searching affixes and Parts-of-Speech for Turkish. The other applications that are mentioned before are working as morphological analysers and concordance tools. Since Turkish is an agglutinative language, affix search is really an important function for researchers, and by using the tagset mentioned above, researchers will be able to search structures such as <V+aor> <V+neg+aor> in order to generate results like *yap+ar yap+ma+z 'as soon as you make this'*. Although the graphs and the lexicon were formed

carefully, ambiguous results or noise in the concordance lines still exist and are waiting for the completion of the on-going disambiguation process.

	Types	Unknowns	Performance
Unpublished written texts	11.037	1.118	89.87
Informative texts	18.278	1.634	91.06
Journals	16.183	1.447	91.06
Newspapers	17.717	1.213	91.15
Spoken texts	9.581	822	91.42
Fictional texts	19.786	1.839	90.71

Table 6: The Performance of the Tagset of NooJ Turkish Module

Acknowledgements

We thankfully acknowledge that this research was supported by a grant from Scientific and Technological Research Council of Turkey (TÜBİTAK Project No. 108K242).

References

Akın, M.D. and A. A. Akın 2007, "An Open Source Natural Language Processing Library for Turkic Languages: Zemberek". *Electrical Engineering*, 431, 38.

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

Göksel, A. and C. Kerslake. 2005, *Turkish : A Comprehensive Grammar*. London & New York: Routledge.

Linden, K., M. Silfverberg, & T. Pirinen 2009, "HFST Tools for Morphology - An Efficient Open-Source Package for Construction of Morphological Analyzers". *Proceedings of the Workshop on Systems and Frameworks for Computational Morphology 2009*. Zürich, Switzerland.

Schmid, H. 2005, "A Programming Language for Finite State Transducers", *Proceedings of the 5th International Workshop on Finite State Methods in Natural Language Processing (FSMNLP 2005)*, Helsinki, Finland.

Silberztein, M. 2003, *NooJ Manual*. 17 September 2011. http://www.nooj4nlp.net
Turkish National Corpus (TNC) Project. http://www.tudd.org.tr / www.tnc.org.tr