Part of speech tagging of Turkish

Ümit Mersinli* – Mustafa Aksan*

1. Introduction

Beginning with Hankamer (1989) and Köksal (1975), studies on computer aided processing of Turkish has emerged following multiple approaches. Turkish, with respect to its complex morphotactics, the in-root phonological alternations forced by harmony rules and other phonological constraints, and the number of homographs, keeps its challenging position in the Natural Language Processing (NLP) literature. After mentioning Oflazer et al. (1994) and Çiçekli (1997) as the applications of 1990s, Akın (2007) and Çöltekin (2010) are two current, rule-based, accessible implementations on the morphological analysis and annotation of Turkish.

This paper presents a root-driven, non-stochastic, graph-based approach to the Parts of Speech Tagging of Turkish. The implemented graphs representing the cascaded finite-state transducer used in modeling Turkish morphotactics is accessible through *www.tudd.org* and are adaptable to other formalisms. The NLP-dictionaries of the module are formed in synchrony with Turkish National Corpus Project[1] and thus, represent the lexicon of present-day Turkish. The architecture of dictionaries and pre-defined lexical features may also contribute to the research on forming the standard NLP-dictionary of Turkish.

1.1. Definitions

In this paper, the term POS Tagging is used in a broad sense that will cover the annotation of inflectional and derivational affixes as well as lexical categories of the base forms. In this respect, the term is reserved for morpheme tagging of Turkish as a preliminary step for morphosyntactic or grammatical tagging as in Treebanks or semantic tagging as in WordNets.

Considering the highly agglutinative nature of Turkish and the amount of homographs, the distinction between inflectional and derivational affixations in Turkish is a challenging issue as also stated in Sezer (2001). In this paper, this distinction is done rather with a computational point of view and thus will not be discussed in detail on a theoretical basis.

* Mersin University.

1 Turkish National Corpus is funded by Scientific and Technological Research Council of Turkiye (TÜBİTAK). (Grant no: 108K242)

1.2. Data

Data of the study are derived from the ongoing Turkish National Corpus (TNC) Project held at Mersin University, Turkey. TNC, as a balanced and representative corpus, is not specifically restricted to any particular subject field, genre or register. Since it contains samples of both written and spoken language, the lexicon and graphs formed for the module represents present-day Turkish.

1.3. Software

Software used for annotation is NooJ as documented in Silberztein (2003). NooJ includes tools for corpus building and management, linguistic analysis, annotation and concordancing. In NooJ "the descriptions of natural languages are formalized as electronic dictionaries, as grammars represented by organized sets of graphs" (Silberztein 2003).

Following the NooJ formalism, Turkish module is also comprised of two basic components; dictionaries and graphs.

2. Dictionaries

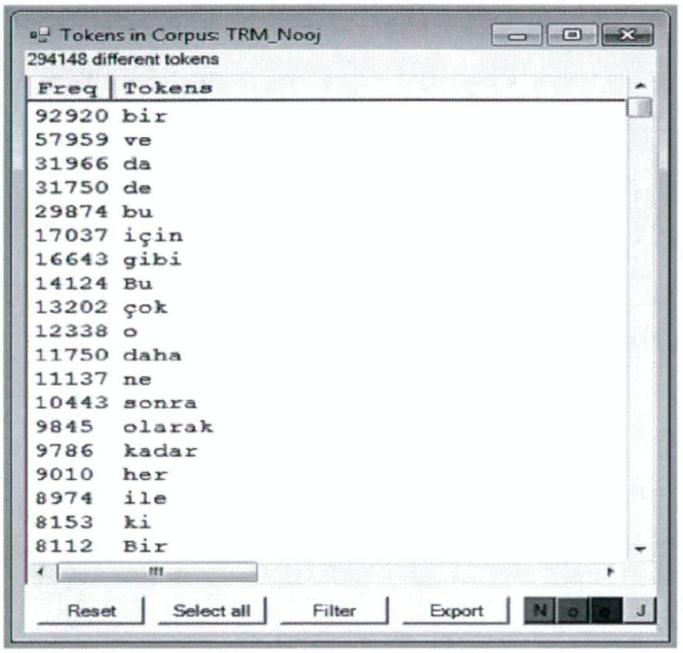
To compile a NooJ dictionary (.nod) file; a Properties Definition file (.def), an Inflectional/Derivational Rule file (.nof) and pre-compiled Raw Dictionaries (.dic) are required (Silberztein 2003). The contents of the three mentioned file formats will be illustrated in the following sections of the study.

2.1. Tokenization

Data for tokenization are extracted from a subcorpus including over 100 texts representing different genres taken from TNC. The subcorpus included over 3,300,000 words forms and over 280,000 tokens when proper nouns, abbreviations and acronyms are excluded.

Below is a sample tokenization with NooJ. Proper Nouns, Acronyms and Abbreviations are filtered out manually from the list of word forms.

Figure 1. Tokenization in NooJ



2.2. Lemmatization

Stemming or affix stripping algorithms and their implementations that can be used for lemmatization in Turkish is out the scope of this study. In this respect, the filtered word forms taken in the previous step are lemmatized manually and an affix database including the affix combinations of Turkish is created. After the pre-tagging process for Lexical Categories, the base form of raw dictionaries is formed as in (1);

(1) al, VB *(take)*
al, AJ *(red)*
yüz, NB *(a hundred)*
yüz, VB *(swim)*
yüz, NN *(face)*

Parts of Speech Tags for Lexical Categories are listed in Table 1.

Table 1. Parts of speech tagset for Turkish.

TAG	POS	EXAMPLE
<VB>	Verb	*git, gel, dur, bak, kal, sus, gör, dök*
<NN>	Noun	*gece, hava, renk, fark, dost, oyun*
<PN>	Pronoun	*bu, kendi, hepsi, herkes, kim, öteki*
<NB>	Number	*iki, üç, beş, sekiz*
<AJ>	Adjective	*mavi, yeni, düz, dürüst, zeki*
<AV>	Adverb	*acaba, asla, bazen*
<PP>	Postposition	*gibi, göre, için, kadar, karşı, rağmen*
<ITJ>	Interjection	*aferin, sağol, haydi, hoşçakal, lütfen*
<CJ>	Conjunction	*ama, çünkü, meğer, üstelik*
<ON>	Onomatopoeia	*takır, vızıl, gürül*
<NP>	Proper Noun	*Atatürk, Mersin, Ümit*
<AB>	Abbreviation, Acronym	*TBMM, TDK*
<MI>	Affirmative particle	*mi, mı, mu, mü*

2.3. Phonemic Alternations

Considering the in-root phonemic alternations as in (2) and (3), phonological rules are defined textually in an Inflectional/Derivational Rule file (phonology_TR.nof) and thus prefixed with "FLX=" as in "FLX=soften_t".

(2) akıl → aklında
mind → mind:GEN+LOC

(3) tıp → tıbbın
medicine → medicine:POSS

The in-root phonemic alternations are listed in Table 3 using the operators in Table 2.

Table 2. Rule operators for NooJ Inflectional/Derivational Grammars.

<B>	delete last character / backspace	<L>	go left
<B2>	delete last two characters	<R>	go right
<D>	duplicate last character	+	OR

Table 3.

tag	rule	example
double	<D>	*af* > *affı*, *zam* > *zamma*
drop	<L><B><R>	*akıl* > *aklını*, *fikir* > *fikrimin*
dropsoften	<B2>b	*kayıp* > *kaybına*, *kutup* > *kutbuna*
compound1	<B>	*anaokulu* > *anaokulları*
compound2	<B2>	*elyazısı* > *elyazıları*, *başağrısı* > *başağrıları*
compound3	<B2>ç	*ipucu* > *ipuçları*
compound4	<B2>k	*ayçiçeği* > *ayçiçekleri*
soften_ch	<B>c	*ağaç* > *ağacı*, *süreç* > *süreci*
soften_k	<B>ğ	*emek* > *emeği*, *diyalog* > *diyaloğu*
soften_p	<B>b	*kitap* > *kitabı*, *mektup* > *mektubu*
soften_t	<B>d	*cilt* > *cilde*, *dört* > *dördünü*
soften_t_er	<B>d + <B>de	*et* > *eder*, *git* > *gider*
soften_t_ar	<B>d + <B>da	*tat* > *tadar*
softendouble	<B>b<D>	*tıp* > *tıbbın*, *muhip* > *muhibbi*
change_an	<B2>an	*ben* > *bana*, *sen* > *sana*
add_er	e	*üz* > *üzer*
add_ar	a	*yap* > *yapar*

2.4. Lexical Features

The Raw Dictionary is compiled with the rules declared in the Inflectional/Derivational Rule file that includes predefined features in the Properties Definition file.

In the current release of the Turkish module, instead of defining lexical properties that can have multiple features as in (4), we preferred to use a binary format for lexical features as in (5).

(4) al,VB+PHON=end_l

(5) al,VB+end_l

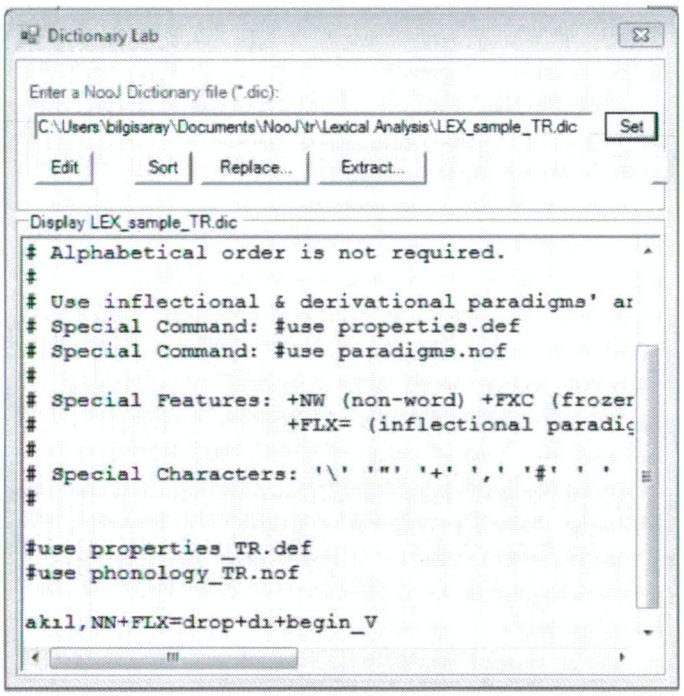
In both of the entries – for use in passivization constraints – it is stated that the Verb "al" ends with consonant "L", whereas in (5) we can also add more phonological features such as "begin_V" (begins with a vowel) for use in duplications like "ev mev". (6) is a sample entry of the final version of raw dictionaries.

(6) akıl,NN+FLX=drop+begin_V

2.5. Compilation

The Raw Dictionaries including a declaration of the related Properties Definition (.def) and Inflectional/Derivational Rule file (.nof) are compiled through the menus Lab → Dictionary as in Figure 1.

Figure 1. Dictionary compilation pane in NooJ.



3. Graphs

After NooJ dictionary files (.nod) are compiled, morphotactics of Turkish is modeled with a Morphological Grammar file (.nom) graphically. NooJ graphs let the user design cascaded finite-state transducers through a graphical interface.

3.1. Overall architecture

Turkish morphological graph is designed to include both derivational and inflectional affixes since, in most cases, the distinction is problematic due to homophonous affixes such as [-mA] serving in both derivational and inflectional processes as in (7) to (12).

(7) saç**ma** (*ridiculous*)
→ adjective forming derivational affix

(8) soruştur**ma** (*investigation*)
→ noun forming derivational affix

(9) git**me** (*don't go* OR *going*)
→ negative OR gerundive

(10) döv**me**li (*must forge* OR *with a tattoo*)
→ part of affix "mAll" OR noun forming derivational affix

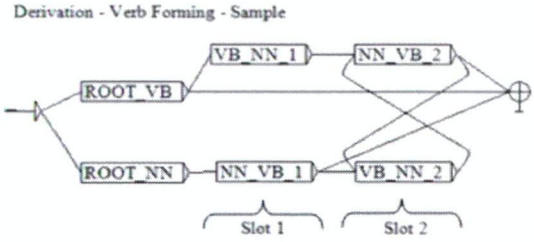
(11) yap**ma**dan (*without doing* OR *from doing*)
→ part of adverbial affix "mAdAn" or gerundive

(12) gide**mememe** (*to my being not able to go*)
→ negative + gerundive + possessive + dative

3.2. Derivation

The derivational subgraph presented in Figure 2 is organized in two slots covering the recursive affixations as in "yaptırttırdı" (s/he ***caused*** *someone to* ***make*** *some other one to* ***get*** *someone to do it).*

Figure 2. Derivational subgraph

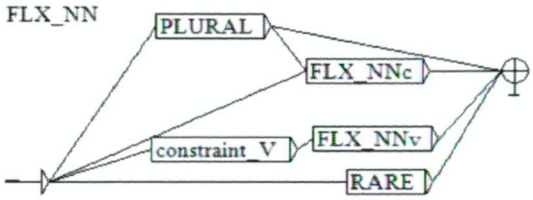


NooJ graphs can be organized in a cascaded manner, so the labels "NN_VB_1" and "NN_VB_2" include the nodes including affixes that derive verbs from nouns.

3.3. Inflection

The inflectional subgraph includes the nominal and verbal inflectional paradigms of Turkish and thus is organized as in two subgraphs. Below is the nominal inflection graph.

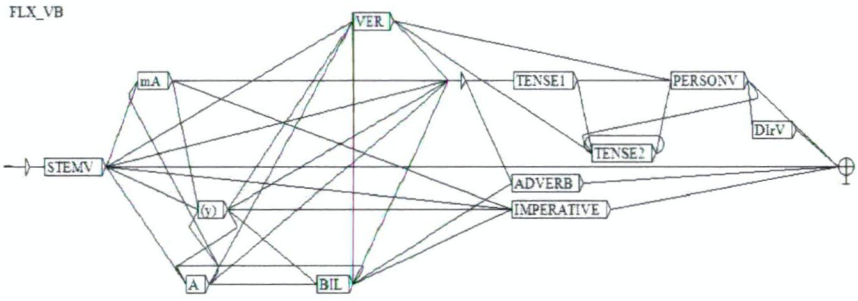
Figure 3. Nominal inflection subgraph



In Figure 3, graphs "FLX_NNc" and "FLX_NNv" are for base forms ending with a consonant and base forms ending with a vowel. This distinction prevents the parser produce artificial ambiguities caused by mostly buffer phonemes.

Figure 4 presents the graph for verbal paradigm.

Figure 4. Finite-state transducer graph for verbal inflection in Turkish



4. Implementation

In the following sections, the performance of the implemented module will be demonstrated.

4. 1. Annotation

Below are sample annotations in (10).

(13) *ölümsüzleştirttirilemeyebileceklerimizdenmişsinizcesine*

öl,VB+(I)m_NN+sIz_AJ+lAş_VB+DIr_VB+t_VB+DIr_VB+(I)l_VB
+A+mA+(y)+A+bil+AcAk_NN+lAr+I+mIz[Poss]+DAn[ABL]
+mIş[Per]+sInIz[2Ppl]+cAsInA_AV

(14) *okulunki buradaki kitaba benzemiyor*

okul,NN+I+n+ki[PN]
bura,NN+DA[LOC]+ki[AJ]
kitap,NN+A[DAT]
benze,VB+mA+yor

(15) *yenilik*
yen,VB+(I)l_VB+Ik_AJ
ye,VB+(I)n_VB+(I)l_VB+Ik_AJ
yeni,AJ+lIk_NN

4.2. Concordancing

Concordancing is done through the "Locate" menu in NooJ. NooJ regular expressions are stated between "<" and ">" symbols to indicate that searching will take place in the text annotation structure.

Below are sample concordance lines for the given search patterns.

<oku,VB>

bir teorik yapının sunduğu yöntemlerle	okuyor	. Kullanım değeri'nin yerine çoktan
bağlantıları kurdurabiliyor muyuz "Raskolnikov"u	okurken	? Kurdurmak zorunda mıyız? "Karanlığın Yüreği
durduran, bir romanın kısaltılmış versiyonunu	okumayı	kendine hakaret olarak algılayan "okur
okumayı kendine hakaret olarak algılayan "	okur	"un klasikten ne umduğunu, ne
zamanı çarşıda olur yakından işitirdik.	Okuma	yazması olmayan biriydi. Nedenini bilemem
bir destan. Eve gidip iyice	okumak	ve anlamak isterdim. Okuduğumda bir
iyice okumak ve anlamak isterdim.	Okuduğumda	bir türlü pazar yerindeki o
Sony Reader PRS-500, elektronik kitap	okumanızı	, RSS haberlerini takip etmenizi ve
seyirlik unsura dönüştürme biçiminde uygulandığını	okuduklarımızdan	öğreniyoruz. Foucault'un "Hapishanenin Doğuşu
zamanı çarşıda olur yakından işitirdik.	Okuma	yazması olmayan biriydi. Nedenini bilemem
bir destan. Eve gidip iyice	okumak	ve anlamak isterdim. Okuduğumda bir
iyice okumak ve anlamak isterdim.	Okuduğumda	bir türlü pazar yerindeki o
bir teorik yapının sunduğu yöntemlerle	okuyor	. Kullanım değeri'nin yerine çoktan
bağlantıları kurdurabiliyor muyuz "Raskolnikov"u	okurken	? Kurdurmak zorunda mıyız? "Karanlığın Yüreği
durduran, bir romanın kısaltılmış versiyonunu	okumayı	kendine hakaret olarak algılayan "okur
okumayı kendine hakaret olarak algılayan "	okur	"un klasikten ne umduğunu, ne
elbette. Hele bir parça Osmanlıca	okuyabilen	varsa tanıdık çevrede, kesin söylüyorum
mi bileceksin? Bakanlıkta şu kadar	okumuş	kişi geceler boyu çalışmış didinmiş
para eder de bir rahmet	okur	bana diye. Ama ben satamam
uzak durmalı. Televizyon seyretmeyim, gazete	okumayım	. İyi de maaş zammına itiraz
ya canım! O kadar biyoloji	okumuşum	, kendi laboratuvarım davar. Olmadı mutfakta
seyirlik unsura dönüştürme biçiminde uygulandığını	okuduklarımızdan	öğreniyoruz. Foucault'un "Hapishanenin Doğuşu
belirtmiştim. "Bir de bu kitabını	oku	!" dercesine Grass'ın kitabını verdiler
kez de ayrıntı abartısıyla karşılaştım.	Okumaya	ara verdim. Gerek şişirilmiş gözlemler
algılama zorluğu çekenler için. Üniversite	okumanın	maliyeti nedir? 22 yaşında okul bitecek
verilere ulaşamıyor. Ürün, saniyede 24 MB	okuyabiliyor	ve 10 MB yazabiliyor. 512MB, 1GB

Figure 5. Sample concordance line 1

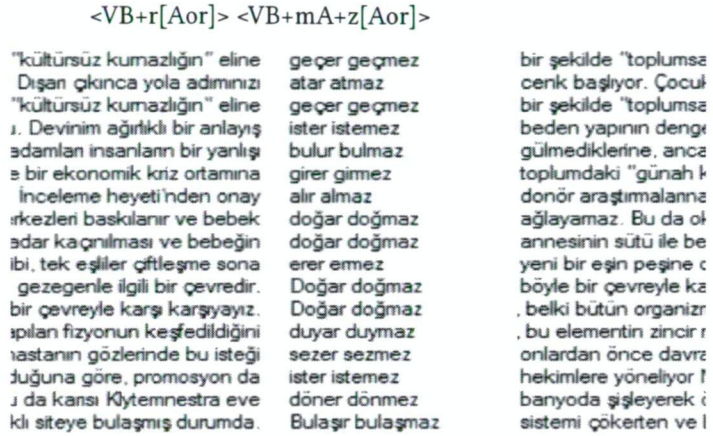


Figure 6. Sample concordance line 2.

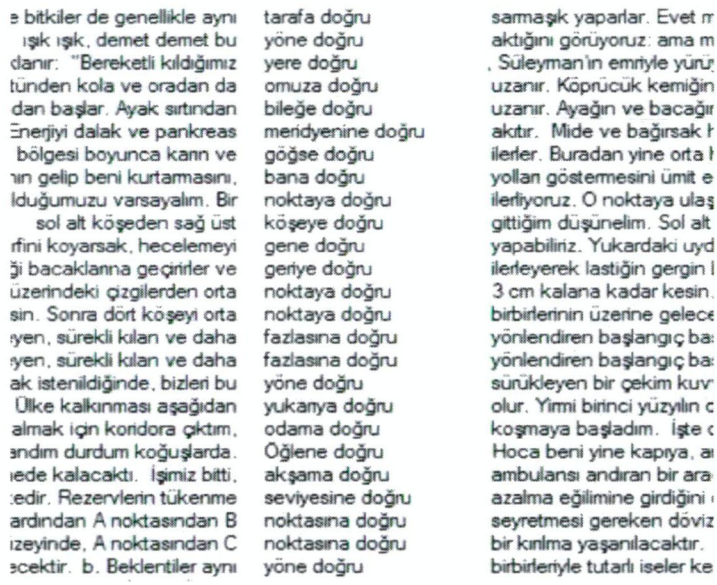


Figure 7. Sample concordance line 3.

5. Conclusion

As presented and demonstrated in this paper, the design of a corpus-driven non-stochastic annotation module for Turkish showed that this preliminary step in Turkish NLP still has unsolved problems and therefore needs further applications. Further areas of NLP such as

information extraction, morphosyntactic annotation or semantic annotation require a full-coverage standard NLP dictionary of Turkish and an accompanying transducer as well.

References

Akın, M. D. & Akın, A. A. 2007. Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi: ZEMBEREK. *Elektrik Mühendisliği* 431, 38.

Çiçekli, İ. & Temizsoy, M. 1997. Automatic creation of a morphological processor in logic programming environment. In: Drogemuller, R. (ed.) *Proceedings of the 5th International Conference on the Practical Application of Prolog (PAP'97).* 22nd-24th April 1997. London, UK. 95–106.

Çöltekin, Ç. 2010. A Freely Available Morphological Analyzer for Turkish. *Proceedings of the 7th International Conference on Language Resources and Evaluation* (LREC2010). Valletta, Malta, May 2010.

Hankamer, J. 1989. Morphological parsing and the lexicon. In: Marslen-Wilson, W. (ed.) *Lexical representation and process* Cambridge, MA: MIT Press. 392–408.

Köksal, A. 1975. *A first approach to a computerized model for the automatic morphological analysis of Turkish.* Ph.D. dissertation. Hacettepe University, Ankara.

Oflazer, K. & Göçmen, E. & Bozşahin, C. 1994. *An Outline of Turkish Morphology.* Technical Report. Middle East Technical University, Ankara.

Sezer, E. 2001. Finite inflection in Turkish. In: Taylan, E. E. (ed.) *The Verb in Turkish.* Amsterdam: Benjamins. 1–47.

Silberztein, M. 2003. *Nooj Manual.* January 10, 2010, from http://www.nooj4nlp.net