A Tool for Tagging Turkish Text

Kemal Oflazer and İlker Kuruöz
Department of Computer Engineering and Information Science
Bilkent University
Bilkent, Ankara 06533 TURKEY
E-mail: {ko,kuruoz}@bilkent.edu.tr
Fax: (0 312) 266-4126

Abstract Automatic text tagging is an important component in higher level analysis of text corpora, and its functionality output can also be used in many natural language processing applications. This paper describes a part-of-speech (POS) tagger for Turkish text. It is based on a full scale two-level morphological specification of Turkish implemented on the PC-KIMMO environment, augmented with statistical information compilation and use, multi-word construct recognition and constraint- and heuristics-based morphological, and POS ambiguity resolution. The tagger also has additional functionality for fine tuning of the morphological analyzer, such as logging erroneous parses, commonly used roots etc. The output of the tagger can be used in further syntactic and semantic analysis.

1 Introduction

As a part of large scale project on computational studies on the Turkish language, we have undertaken development of a number of tools for analyzing Turkish text. This paper describes one such tool – a text tagger – for Turkish. The tagger is based on a full scale two-level morphological specification of Turkish implemented on the PC-KIMMO environment [1, 12] and represents substantial improvement over our previous work [11]. In this paper we describe the functionality of our tagging application along with various techniques that we have employed to deal with various sources of ambiguities.

2 Tagging Text

Automatic text tagging is an important step in discovering the linguistic structure of large text corpora. Basic tagging involves annotating the words in a given text with various pieces of information, such as part-of-speech and other lexical features. Part-of-speech tagging facilitates higher-level analysis, such as syntactic parsing, essentially by performing some ambiguity resolution using relatively cheaper methods.

The most important functionality of a tagger is the resolution of the parts-of-speech of the lexical items in the text. This, however, is not a very trivial task since many words are in general ambiguous in their part-of-speech, for various reasons. In English, for example a word such as *make* can be verb or a noun. In Turkish even though there are ambiguities of such sort, the agglutinative nature

of the language usually helps resolution of such ambiguities due to morphotactical restrictions. On the other hand, this very nature introduces another kind of ambiguity, where a lexical form can be morphologically interpreted in many ways. For example, the word *yüzer*, can be broken down as:[1,2]

	yüzer	**POS**	**English**
1.	V(yüz)+AOR+3SG	V	he swims
2.	V(yüz)+VtoADJ(er)	ADJ	floating

There are two readings of *yüzer*, one as a verb and the other as an adjective. However, when a word appears in the context of other words, the ambiguity is often resolvable: in

Ahmet'i yüzer evde dinlenirken bulduk

the word *yüzer* can only be an adjective meaning (floating), since it is a part of the compound noun phrase *yüzer evde*. As a more complex example we can give the following:

	alınmış	**POS**	**English**
1.	ADJ(al)+2SG-POSS+NtoV()+NARR+3SG[3]	V	(it) was your red (one)
2.	ADJ(al)+GEN+NtoV()+NARR+3SG	V	(it) belongs to the red (one)
3.	N(alın)+NtoV()+NARR+3SG	V	(it) was a forehead
4.	V(al)+PASS+VtoAdj(mis)	ADJ	(a) taken (object)
5.	V(al)+PASS+NARR+3SG	V	(it) was taken
6.	V(alın)+VtoAdj(mis)	ADJ	(an) offended (person)
7.	V(alın)+NARR+3SG	V	(s/he) was offended

It can be seen that it is in general rather hard to select one of these interpretations without doing substantial analysis of the local context and even then one can not fully resolve such ambiguities.

The functionality offered by a tagger are also important to higher level analysis of text such as syntactic or semantic analysis. Most lexical and morphological ambiguity problems that come up in such processing can be eliminated by local analyses in the tagger whose complexity is less than full-fledged parsing. An additional problem that can be off-loaded to the tagger is the recognition of multi-word constructs. In Turkish, such a recognizer can recognize multi-word constructs, like *koşa koşa* or *yapar yapmaz*, where both components are verbal but the compound construct is a manner or temporal adverb,[4] instead of a parser dealing with them at the syntactic level. Turkish abounds with such forms. Furthermore it is also possible to recognize various proper nouns, and compound nouns with this functionality. Such help from an underlying functionality would make the development of parsers for Turkish (such as [5, 7]) considerably easy.

Researchers have used a number of different approaches for building text taggers. Karllson [8] has used a rule-based approach where the central idea is to maximize the use of morphological information. Local constraints expressed as rules basically discard many alternatives parses whenever possible. Brill [2] has designed a rule-based tagger for English. The tagger works by automatically recognizing rules and remedying its weaknesses, thereby incrementally improving its performance. More recently, Koskenniemmi et. al., [10, 14] used a rule-based approach implemented with finite-state machines.

[1] Output of the morphological analyzer is edited for clarity.

[2] We are also ignoring that *yüz* as a verb has two different meanings.

[3] In Turkish, all adjectives can be used as nouns, hence with very minor differences adjectives have the same morphotactics as nouns.

[4] Though for the latter case, the form *olur olmaz* may also have an adjectival reading.

A completely different approach to tagging uses statistical methods, such as Hidden Markov Models. (e.g., DeRose [6], Cutting et.al., [4], Church [3]). These systems essentially train a Hidden Markov Model using previously hand-tagged corpus and provide the capability of resolving ambiguity on the basis of most likely interpretation. The models that have been widely used assume that the part-of-speech of a word depends on the categories of the two preceding words.

2.1 An example

We can describe the process of tagging by showing the analysis for sentence,

> *İşten döner dönmez evimizin yakınında bulunan derin göldeki yüzer evde dinlenmek en büyük zevkimdi.*

which we assume has been processed by the morphological analyzer with the following output:

	işten	**POS**
1.	N(iş)+ABL	N*
	döner	
1.	N(döner)	N
2.	V(dön)+AOR+3SG	V
3.	V(dön)+VtoAdj(er)	ADJ*
	dönmez	
1.	V(dön)+NEG+AOR+3SG	V*
2.	V(dön)+VtoAdj(mez)	ADJ
	evimizin	
1.	N(ev)+1PL–POSS+GEN	N*
	yakınında	
1.	ADJ(yakın)+3SG–POSS+LOC	N*
2.	ADJ(yakın)+2SG–POSS+LOC	N
	bulunan	
1.	V(bul)+PASS+VtoADJ(yan)	ADJ
2.	V(bulun)+VtoADJ(yan)	ADJ*
	derin	
1.	N(deri)+2SG–POSS	N
2.	ADJ(derin)	ADJ*
3.	V(der)+IMP+2PL	V
4.	V(de)+VtoADJ(er)+2SG–POSS	N[5]
5.	V(de)+VtoADJ(er)+GEN	N

	göldeki	**POS**
1.	N(göl)+LOC+NtoADJ(ki)	ADJ*
	yüzer	
1.	V(yüz)+AOR+3SG	V
2.	V(yüz)+VtoADJ(er)	ADJ*
3.	ADJ(yüzer)	ADJ
	evde	
1.	N(ev)+LOC	N
	dinlenmek	
1.	V(dinle)+PASS+VtoINF(mak)	V
2.	V(dinlen)+VtoINF(mak)	V*
	en	
1.	N(en)	N
2.	ADV(en)	ADV*
	büyük	
1.	ADJ(büyük)	ADJ*
	zevkimdi	
1.	N(zevk)+1SG–POSS+ NtoV()+PAST+3SG	V*

There are 1440 possible choices of tags for this sentence although almost all except one give rise to ungrammatical or implausible sentence structures. The correct choices of tags are marked with *. There are number of points that are of interest here:

- the construct *döner dönmez* is actually a temporal adverb meaning ... *as soon as .. return(s)* ... hence these two lexical items can be coalesced into a single lexical item and tagged as a temporal adverb.

[5] Although, the final category is adjective the use of possessive (and/or case, number) suffixes indicate nominal usage, as any adjective in Turkish can be used as a noun

- The second person singular possessive interpretation of *yakınında* is not possible since this word forms a simple compound noun phrase with the previous lexical item and the third person singular possessive functions as the compound marker.
- The word *derin (deep)* is the modifier of a simple compound noun *derin göl (deep lake)* hence the second choice can safely be selected. The verbal root in the third interpretation is very unlikely to be used in text, let alone in second person imperative form. The fourth and the fifth interpretations are not plausible adjectives from aorist verbal forms almost never take any further inflectional suffixes. The first interpretation (meaning *your skin*) may be a possible choice but can be discarded in the middle of a longer compound noun phrase.
- The word *yüzer* also has to be interpreted as an adjective since in that case it will be a part of the compound noun *derin göldeki yüzer ev.*
- The word *en* preceding an adjective indicates a superlative construction and hence the noun reading can be discarded.
- On the other hand the ambiguity of the word *dinlenmek* is a semantic ambiguity and can be resolved with semantic analysis and not in the tagger. The first reading corresponds to *to be listened to* and the second reading corresponds to *to rest.* Most likely the words *göl (lake)* and *zevk (pleasure)* would be help select with the latter reading of this word, during semantic analysis.

The tagger should essentially reduce the possible parses to the minimum possible, employing usage and other statistical information and various constraint rules and heuristics.

3 The Tagging Tool

The tagging tool for Turkish integrates various functionality discussed above with a user interface as shown in Figure 1.

The tagger contains two text windows, one for the text being tagged, and the other for displaying tagged text. It will request user assistance whenever it can not resolve any ambiguity using its current sources of information.

The user has the option of selecting the scope of word usage frequency statistics. By default, the tagger uses a cumulative global statistic that has been collected during tagging of other similar texts. If, however local statistics are preferred, the tagger starts compiling local frequencies and statistical decisions just rely on this local information.

It is also possible to update the rule base used in the processing of multi-word constructs, for example when an unknown proper noun encountered. In this case the specifications can be modified and changes are made effective by reloading.

The user has the chance of analyzing specific word groups by setting *skip parses less than* option. This option enables user to compile local statistics about words having more than a specific number of parses.

If an ambiguity can not be resolved, all parses of the word are displayed and user is asked to choose the correct parse and/or indicate any erroneous parses. The user can configure the tagger to mark

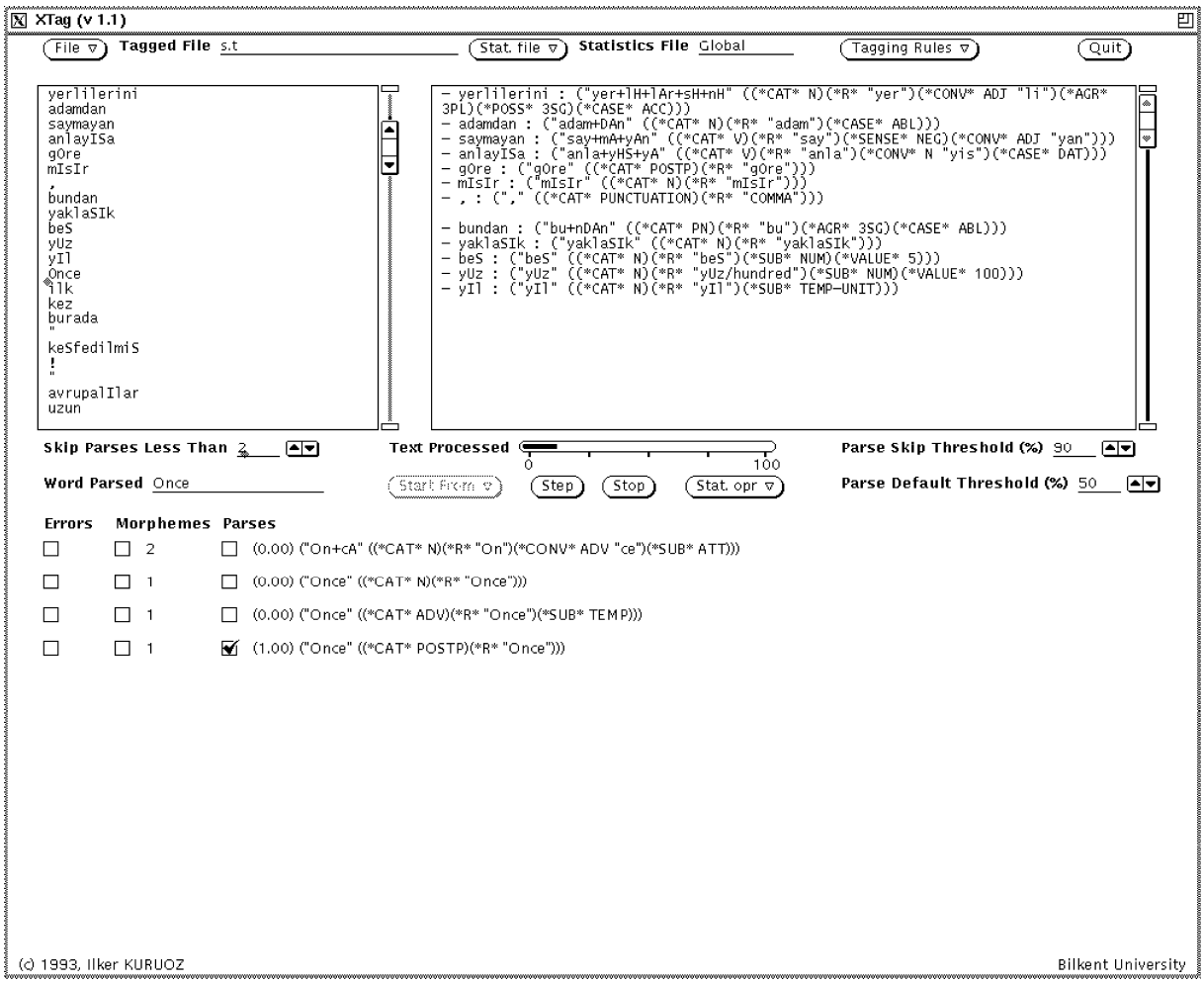

Figure 1: User interface of tagging tool

as the default selection, the parse that has been selected over a percentage *parse default threshold* according to the statistics. This selection is not processed if the user does not confirm. However, by setting the *parse skip threshold* user can ignore confirmation on words which have been encountered certain number of times. The word is tagged with the parse having highest selection rate beyond *parse skip threshold.*

The tool keeps a log during tagging process, so that user can stop tagging at any point in the text and continue later without losing any information.

As mentioned the tagger also compiles local or global word usage frequency statistics, depending on user preferences. This statistical information is used in ambiguity resolutions A typical use for such information is in weeding out very infrequently used or colloquial root words from the lexicon. For example, word *çözüm* has the following three morphological parses:

	çözüm	**POS**	**English**
1.	N(çöz)+1SG–POSS	N	my intestines
2.	N(çöz)+NtoV()+PR-CONT+1SG	V	(I am) intestines
3.	N(çözüm)	N	solution

In all occurrences of this word in a text, the correct reading is the third one as the first two readings are either not to be found in current usage or non-sense, although they are morphologically perfectly acceptable.

Another similar example is the word *çimdik* whose parses are:

	çimdik	**POS**	**English**
1.	N(çim)+NtoV()+PAST+1PL	V	(we were) grass
2.	N(çimdik)	N	pinch
3.	V(çim)+PAST+1PL	N	we swam

Here the third reading is based on a root that is used very infrequently and only then in certain local dialects. The first is very implausible although the root *çim (grass)* is very common. The second reading is always the correct reading.

In addition to compilation of such statistics the tagger also logs any erroneous parses indicated by the user so that the morphological analyzer can be fixed or fine-tuned.

3.1 The Multi-word Construct Processor

As mentioned before, tagging text on lexical item basis may generate spurious or incorrect results when multiple lexical items act as single syntactic or semantic entity. For example, in the sentence

Şirin mi şirin bir köpek koşa koşa geldi. (A very cute dog came running.)

the fragment *şirin mi şirin* constitutes a duplicated emphatic adjective in which there is an embedded question suffix *mi* (written separately in Turkish), and the fragment *koşa koşa* is a duplicated verbal construction where each form has the morphological parse:

	koşa		**English**
1.	N(koşa)	N	twin
2.	V(koş)+OPT+3SG	V	let him run

but yet the duplicated form has the grammatical role of manner adverb in the sentence. The purpose of the multi-word construct processor is to detect and tag such constructs in addition to various other semantically coalesced forms such as proper nouns, etc.

Following is a set of multi-word constructs that we handle in our tagger. This list is not meant to be comprehensive, and new construct specifications can easily be added to our system.

1. duplicated optative and 3SG verbal forms functioning as manner adverb, e.g., *koşa koşa*,
2. aorist verbal forms with root duplications and sense negation functioning as temporal adverbs, e.g.,*yapar yapmaz*,
3. duplicated verbal and derived adverbial forms with the same verbal root acting as temporal adverbs, e.g.,*gitti gideli*,
4. duplicated compound nominal form constructions that act as adjectives, e.g., *güzeller güzeli*,
5. adjective or noun duplications that act as manner adverbs, e.g., *hızlı hızlı*, *ev ev*,
6. emphatic adjectival forms involving the question suffix, e.g., *güzel mi güzel*,

7. word sequences with specific usage whose semantics is not compositional, e.g., *yanı sıra, hiç olmazsa*,
8. proper nouns, e.g., *Süleyman Demirel,Topkapı Sarayı*,
9. idiomatic forms which are never used by themselves, e.g., *gürül gürül*,
10. other idiomatic forms.

3.1.1 Specifying Multi-word Constructs

Most multi-word constructs require recognition of a sequence of forms with certain roots and/or morphological features. For example, all forms of the second type above have the morphological structure:

V(root)+AOR+3SG and V(root)+NEG+AOR+3SG

or all forms of the third type have the morphological structure

V(root)+PAST+PERSON/NUMBER and V(root)+VtoADV(yali)

where in both cases both roots have to be the same. In order to specify such constructs we have developed a simple specification format whereby one can specify rules describing these multi-word constructs. Such rules can then be added to the system to extend its functionality as new forms are required.

Our pattern specification provides for the specification of:

1. one or more morphological patterns involving constraints for morphological features, such as *root, root category, agreement, case, aspect, final category*, etc., describing the constraints that has to be satisfied by each sequential lexical form.
2. An output pattern that describes the format of what is to be output if the pattern is matched.
3. Variables referring to the roots (R) or the lexical forms (W) of the words involved, or a don't care symbol.

Morphological constraints, rules and the output pattern are separated by suitable punctuation: '=' is used in the constraints, ',' separates the features within a rule, ';' separates rules and ':' separates the output pattern.

For example the constructs like *yapar yapmaz* are specified by:

```
Root = R1 , Cat = V , Aspect = AOR , Agr = 3SG ;
Root = R1 , Cat = V , Aspect = AOR , Agr = 3SG , Sense = NEG:
Output = ((*CAT* ADV)(*R* "W1 W2")).[6]
```

On the other hand, certain word sequences with specific usage, e.g.,*yanı sıra* can be specified using a number of rules that have to be consulted in the given sequence:

[6]The output of the morphological analyzer is actually a *feature-value* list in the standard LISP format.

```
Root = bizler, Case = GEN ;
Root = yan , Cat = N , Poss = 3SG ;
Root = sıra , Cat = N ;
Output = ((*CAT* POSTP)(*R* "W1 W2 W3")).

Root = sizler, Case = GEN ;
Root = yan , Cat = N , Poss = 3SG ;
Root = sıra , Cat = N ;
Output = ((*CAT* POSTP)(*R* "W1 W2 W3")).

Root = onlar, Case = GEN ;
Root = yan , Cat = N , Poss = 3SG ;
Root = sıra , Cat = N ;
Output = ((*CAT* POSTP)(*R* "W1 W2 W3")).

Root = ?, Agr = A1, CASE = GEN;
Root = yan , Cat = N , Poss = A1 ;
Root = sıra , Cat = N ;
Output = ((*CAT* POSTP)(*R* "W1 W2 W3")).

Root = ?, CASE = GEN;
Root = yan , Cat = N , Poss = 3SG ;
Root = sıra , Cat = N ;
Output = ((*CAT* POSTP)(*R* "W1 W2 W3")).

Root = yan , Cat = N , Poss = A1 ;
Root = sıra , Cat = N ;
Output = ((*CAT* POSTP)(*R* "W1 W2")).
```

The rules above cover possible formations involving *yanı sıra* because *yanı* can be a part of a compound noun, hence has to agree in certain aspects with the preceding lexical forms.

Multi-word constructs that just involve a sequence of patterns with no constraints on any of the forms can also be similarly described by specifying a sequence of roots. In most cases especially when proper nouns are involved the inflections of the last form in the sequence have to be inherited by the multi-word construct. This is indicated at the output by a '$' so that unspecified features are inherited from the last form. For example the form *Topkapı Sarayı'nın (Topkapı Palace+GEN)* would be matched by a rule.

```
Root = Topkapı;
Root = Sarayı:[7]
Output = ((*CAT* N)(*R* "Topkapı Sarayı")(*SUB* PROP)$).
```

which would then generate a form such as:

((*CAT* N)(*R* "Topkapı Sarayı")(*SUB* PROP)(*CASE* GEN))

[7] Note here that we are treating the whole word as a single root and not as the noun *saray* followed by a compound marker.

3.2 Employing constraints for morphological ambiguity resolution

Morphological analysis does not have access to syntactic context. When the morphological ambiguity of a lexical form which has several distinct morphological analyses can not be resolved automatically using statistical information, one may have to resort to using local syntactic context to resolve ambiguities. Voutilainen and Heikkila [13] have proposed a *constraint grammar* approach where one specifies constraints on the local context of a word to disambiguate among multiple readings of a word. Their approach has, however, been applied to English where morphological information has almost no use in such resolution.

One of the very important applications of the use of such syntactic context is the syntactic recognition of compound noun phrases (not necessarily the resolution of the internal structure of such phrases). In compound noun phrases, even though each lexical form may have a number morphological parses, structural constraints work together so that all except one or two combinations of possible readings can be eliminated. As an exaggerated example we can give the following compound noun phrase:

> *evin çürümüş tahta duvarlarının kavlayan soluk renkli boyaları*
> (the pale colored paint peeling off the rotting wooden walls of the house)

The individual morphological parses of each of the words in this phrase can be given as:

	evin	**POS**
1.	N(ev)+2SG-POSS	N
2.	N(ev)+GEN	N
3	N(evin)	N
	çürümüş	
1.	V(çürü)+VtoADJ(miş)	ADJ
2.	V(çürü)+NARR+3SG	V
	tahta	
1.	N(taht) + DAT	N
2.	N(tahta)	N(ADJ)
	duvarlarının	
1.	N(duvar)+3PL+3SG-POSS+GEN	N
2.	N(duvar)+3PL+2SG-POSS+GEN	N
3.	N(duvar)+ 3PL-POSS + GEN	N
4.	N(duvar)+3PL+ 3PL-POSS + GEN	N

	kavlayan	**POS**
1.	V(kavla)+VtoADJ(yan)	ADJ
	soluk	
1.	N(soluk)	N
2.	ADJ(soluk)	ADJ
	renkli	
1.	N(renk)+NtoADJ(li)	ADJ
	boyası	
1.	N(boya)+3SG-POSS	N

By using the heuristics that in the absence of any punctuation, the longest sequence of lexical forms that match a certain pattern (defined by a regular expression of feature structures) makes up a compound noun, we can resolve the ambiguities in the phrase above as follows:

1. The genitive case in *evin* signals an owner qualifier in a noun phrase and hence the beginning of compound noun.

2. After an owner qualifier one or more property qualifier may come. In this case the adjective reading of *çürümüş* can be taken as the qualifier. Note that it is also possible to select as the first of third readings of *evin* which are in nominative case and hence can function as the subject of the the verbal interpretation of this word. This would however require that the sentence or a coordinating phrase end there which would be indicated by some punctuation.

3. Qualifiers may not be in dative case (unless they are related to embedded participle phrases which we do not consider) hence the second reading of *tahta* is chosen. (This reading actually functions as an adjective since nouns denoting materials can be used as adjectives.)

4. Since the owner qualifier *evin* is singular and has an 3SG agreement (not explicitly marked) the first reading of *duvarlarının* has to be chosen the others can be safely discarded. Note that, this now signals another owner qualifier with a plural agreement.

5. The adjective reading of *soluk* can be preferred over the noun reading since we are still in the middle of a noun phrase and there is a pending possessor (*duvarlarının*) which has not been matched with a possessed nominal form.

In this way, we can discard all but one reading of this sequence of words by making one left-to-right pass.

There are a number of other constraints that can be used in various contexts. Some of them are:

1. The form *en* is a superlative marker and also a noun. When it is followed by an adjective, the superlative marker reading can safely be selected.

2. Derived adjectival readings of the aorist forms of verb almost never take any further inflectional suffixes (see *derin* in the earlier example). Hence such readings can be discarded in such cases.

3. Temporal nouns ending with the copula *+dir* have a verbal reading in a sentence final position, but has a temporal adverb reading otherwise.

4. Adverbial readings before nouns can be discarded.

5. Certain features being the same, cohorts with smaller number of morphemes (especially forms not having derivational suffixes) can be preferred. An example of this is the word *kaplamalarda* which can be analyzed as:

   kaplamalarda

   1. N(kaplama)+3PL+LOC on the linings
   2. V(kapla)+VtoN(ma)+3PL+LOC on the (act of) covering

   In this case, it is possible to select the the first reading as it has a more specific semantics.

4 Further Improvements

Only word usage frequency statistics have been used for disambiguation so far, however higher level of statistical models can be employed. Cutting et.al. [4] have recently used such a method based on Hidden Markov Modeling and we intend to augment our tagger with the functionality provided by such statistical models.

Currently, the speed of the tagger is limited by that of the morphological analyzer, but we have ported the morphological analyzer to the XEROX PARC system developed by Karttunen and Beesley [9], which can analyze Turkish word forms at about 500 forms/sec on workstations.

5 Conclusions

This paper has presented an overview of a tool for tagging Turkish text along with various issues that have come up in disambiguating among morphological parsing of Turkish words. The tool also provides for recognition of multi-word constructs that behave a single syntactic and semantic entity in higher level analysis. Although the tool provides for compilation of certain statistical information, it does not yet employ more sophisticated statistical methods like Hidden Markov Models. We expect to integrate such approaches in the future.

6 Acknowledgements

This work was done as a part of a large scale NLP project (TU-LANGUAGE) funded by a NATO Grant under the Science for Stability Program.

References

[1] E. L. Antworth, *PC-KIMMO: A Two-level Processor for Morphological Analysis.* Summer Institute of Linguistics, Dallas, Texas, 1990.

[2] E. Brill, "A simple rule-based part-of-speech tagger," in *Proceedings of the Third Conference on Applied Computational Linguistics*, Trento, Italy, 1992.

[3] K. W. Church, "A stochastic parts program and noun phrase parser for unrestricted text," in *Proceedings of the Second Conference on Applied Natural Language Processing (ACL)*, pp. 136–143, 1988.

[4] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun,. "A practical part-of-speech tagger,". Technical report, Xerox Palo Alto Research Center, 1993.

[5] C. Demir, "An ATN grammar for Turkish," M.Sc. thesis, Department of Computer Engineering and Information Sciences, Bilkent University, Ankara, Turkey, July 1993.

[6] S. J. DeRose, "Grammatical category disambiguation by statistical optimization," *Computational Linguistics*, vol. 14, pp. 31–39, 1988.

[7] Z. Güngördü, "A lexical-functional grammar for Turkish," M.Sc. thesis, Department of Computer Engineering and Information Sciences, Bilkent University, Ankara, Turkey, July 1993.

[8] F. Karlson, "Constraint grammar as a framework for parsing running text," in *Proceedings of COLING– 90, the 13th International Conference on Computational Linguistics*, volume 3, pp. 168–173, Helsinki, Finland, 1990.

[9] L. Karttunen and K. R. Beesley,. "Two-level rule compiler,". Technical Report, XEROX Palo Alto Research Center, 1992.

[10] K. Koskeniemi, P. Tapanainen, and A. Voutilainen, "Compiling and using finite-state syntactic rules," in *Proceedings of COLING–92, the 14th International Conference on Computational Linguistics*, volume 1, pp. 156–162, Nantes, France, 1992.

[11] K. Oflazer, "A text tagger for Turkish," in *Proceedings of Second Turkish Symposium on Artificial Intelligence and Neural Networks*. Bosphorus University, 1993.

[12] K. Oflazer, "Two-level description of Turkish morphology," in *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, April 1993. A full version is to appear in *Literary and Linguistic Computing*, Vol.9 No.2, 1994.

[13] A. Voutilainen, J. Heikkila, and A. Anttila, *Constraint Grammar of English.* University of Helsinki, 1992.

[14] A. Voutilainen and P. Tapanainen, "Ambiguity resolution in a reductionist parser," in *Proceedings of EACL'93*, Utrecht, Holland, 1993.