A Freely Available Morphological Analyzer for Turkish

Çağrı Çöltekin

Center for Language and Cognition (CLCG)
University of Groningen
c.coltekin@rug.nl

Abstract

This paper presents TRmorph, a two-level morphological analyzer for Turkish. TRmorph is a fairly complete and accurate morphological analyzer for Turkish. However, strength of TRmorph is neither in its performance, nor in its novelty. The main feature of this analyzer is its availability. It has completely been implemented using freely available tools and resources, and the two-level description is also distributed with a license that allows others to use and modify it freely for different applications. To our knowledge, TRmorph is the first freely available morphological analyzer for Turkish. This makes TRmorph particularly suitable for applications where the analyzer has to be changed in some way, or as a starting point for morphological analyzers for similar languages. TRmorph's specification of Turkish morphology is relatively complete, and it is distributed with a large lexicon. Along with the description of how the analyzer is implemented, this paper provides an evaluation of the analyzer on two large corpora.

1. Introduction

Morphological analysis is an important part of many computational linguistics applications. It is particularly important for morphologically complex languages, where some of the linguistic information expressed by multiple words and the relations between the words in other languages are confined into single words. Regardless of the morphological complexity of the language being processed, morphological analysis helps natural language processing systems by reducing lexicon size and effects of data sparseness.

The morphological analyzers in use today are generally rule-based systems, typically implemented using finite state transducers (FSTs). The time and effort necessary for developing a finite state morphological analyzer depends on the complexity of the morphology of the language as well as the available resources—such as grammars or testing data. The implementation of a morphological analyzer typically requires weeks, or more likely, months of expert effort.

The rules used in a morphological analyzer are more or less invariant of the purpose the morphological analyzer is used for. However, different applications require slight modifications for various reasons. For example, one may wish to have less strict rules while processing spoken language data, but more strict if the analyzer is used for a spell checker. Similarly, one may want to adapt the analyzer to a dialect other than the standard one, or to a similar language. Or, we may want to have a finer grained classification of a certain part of speech class or a certain affix. All these applications require modification of the rule set or the lexicon in certain ways.

Mostly due to licensing problems with the tools and resources, not all morphological analyzers found in the literature are freely or easily accessible. Even if developers of an analyzer provide some limited access, as long as the morphological specification and the lexicon is not distributed or the tools needed for using the analyzer have restriction on their use; these restrictions create inconveniences to the users, and in some cases they make it impossible to use the analyzer. Hence, causing duplication of the labor. Besides user convenience, a freely available morphological analyzer used by a wider user base would result in a better tested and maintained system. This paper intends to address these problems by presenting a freely available morphological analyzer for Turkish.

Development of TRmorph had been initiated due to one of the reasons listed above: in a study (Çöltekin and Bozsahin, 2007) where morphological analysis and segmentation of corpora of child directed speech in CHILDES database (MacWhinney and Snow, 1990) was necessary. However, it has evolved into an analyzer with a fairly complete rule set and lexicon, and the version presented here is developed to analyze modern standard written Turkish. We believe it can be a valuable resource especially for researchers who would need to adapt it for a particular use, or similar languages.

The analyzer reported here is not the first morphological analyzer for Turkish. Early attempts date back to Hankamer (1986), and a well-known two-level analyzer was developed by Oflazer (1994). However, the analyzer presented here, to our knowledge, is the first freely available morphological analyzer for Turkish. As well as the two-level specification, the tools and resources used for developing the analyzer are distributed under free licenses.

The two-level specification described in this paper is freely available at http://www.let.rug.nl/coltekin/trmorph/ and distributed under the GNU General Public License (GPL).[1] As well as the two-level implementation, the distribution includes a relatively large lexicon adapted from a free spell checker for Turkish, Zemberek (Akın and Akın, 2007). TRmorph is completely implemented using freely available Stuttgart finite state transducer tools (SFST).

The SFST (Schmid, 2005) is a freely available finite state tool set particularly aimed for implementing morphological analyzers. It uses a simple specification language mainly based on regular expressions, with additions of the well known two-level operators (Koskenniemi, 1983; Karttunen and Beesley, 2005) that are particularly useful in implementing phonological (or orthographic) alternations. SFST has been used to implement morphological analyzers for a number of other languages with differing morphological

[1] http://www.gnu.org/licenses/gpl.html

complexity including German (Helmut Schmid and Heid, 2004), Italian (Zanchetta and Baroni, 2005), and Finnish (Pirinen, 2008).
Due to lack of tools that work with non-ASCII character sets, most NLP tools for Turkish use capital letters to represent Turkish letters missing in ASCII. Since most corpora today are in character sets that can represent all the Turkish characters, this method causes some inconvenience, especially if upper case and lower case difference is significant for the application at hand. The tool set used by TRmorph, SFST, can use UTF-8. So, another, albeit minor, feature of TRmorph is the use of UTF-8 encoding which is more suitable for modern corpora.
The next section presents a brief overview of Turkish morphology. Section 3. describes the implementation of the morpho-phonological rules in TRmorph, followed by an evaluation of the analyzer in Section 4.. Section 5. concludes after a discussion of limitations and future work in Section 3.4..

2. Turkish Morphology

Turkish is an agglutinating language with relatively complex morphology. We will give a brief description to demonstrate the complexity of the task in this section. We will also try to provide relevant information on the morphology of the language in Section 3. as much as the space limitations allow. Comprehensive descriptions can be found in Turkish grammars, such as Göksel and Kerslake (2005), Kornfilt (1997) or Lewis (2000).
Turkish words can be formed by a potentially long concatenation of morphemes. A widely used example is given in example (1) below.

(1) İstanbul-lu-laş-tır-ama-dık-lar-ımız-dan-mış-sınız
'You are (supposedly) one of those who we could not convert to an İstanbulite'

Even though this example is made for demonstration and somewhat difficult to find in real language use, it is perfectly intelligible and one may even stretch more and come up with longer sequences of morphemes. Furthermore, long sequences of morphemes in real-world data is not uncommon. Based on TRmorph's analyses the morpheme sequences of length 5 or more ranges between 6 to 10% (see Section 4. for more detail).
As well as possibly long sequences of morphemes, some suffixes attach to the stems recursively. Example (2) demonstrates this with suffix `-ki`.

(2) ev -de **-ki** -nin **-ki** -ler -de **-ki** ...
house -LOC -REL -POS3s -REL -PLU -LOC -REL ...

Multiple usage of this suffix is rare, however, there is no principled reason to put an arbitrary limit. Similar phenomena may also occur with causative, and arguably with some verbal suffixes that form compound verbs. In the corpora used for evaluation, maximum number of `-ki` suffixes in a single word is 2, and maximum number of causatives is 3.
Except a few borrowed derivational prefixes, Turkish is a suffixing language. Derivational suffixes generally attach to root forms, and they are not as productive as inflectional morphology. However, there are exceptions to this generalization, and there are cases where the same derivational suffix repeatedly attaches to the same stem (e.g. `göz-lük-çü-lük`). Except a few very productive derivational affixes, TRmorph primarily deals with inflectional morphology. A comprehensive inventory of derivational suffixes is included in the TRmorph distribution and can optionally be used for analysis.
Together with relatively complex morphotactics, Turkish also has a number of morpho-phonological alternations, such as vowel harmony and consonant (de)voicing. TRmorph implements these alternations, including some exceptions, and we will go through these alternations in Section 3..
Both the morphotactics, and the morpho-phonological rules of the language are relatively regular. Nevertheless, there are a number of exceptions that we will list in more detail in the next section.

3. Two-level implementation

TRmorph is implemented using the Stuttgart finite state transducer tools. Like many other finite state tools for morphological analysis, the specification of morphology in SFST consists three major parts: a finite state machine (FSA), specified using regular expressions, for morphotactics; a set of two-level rules for specifying phonological or orthographic alternations; and a lexicon listing the root forms of the words. The lexicon stores the class of each root word, and some lexical irregularities.
In most part TRmorph follows the morphological description in a recent grammar of Turkish (Göksel and Kerslake, 2005). As well as hand-made examples, the analyzer is tested on a word list collected from on-line newspapers throughout its development. To have a better coverage of the corpus, there has been a number of divergences from grammar book specification. However, the divergences do not necessarily mean diverging from the standard language. TRmorph, as distributed in the url specified above, is designed to work for standard written Turkish, and follows it closely.

3.1. The lexicon

The TRmorph distribution comes with two alternative lexicons. A small lexicon which is created during the implementation, and a relatively large lexicon adopted from the lexicon of the Zemberek project with a large number of corrections and modifications. The former lexicon is checked carefully and it is relatively error-free, but the latter one, despite inaccuracies, provides a more reasonable coverage. The lexicons contain 1500 and 37101 words respectively. Both lexicons classify the lexical items into 9 categories: adjectives, adverbs, conjunctions, interjections, nouns, postpositions, pronouns, proper names and verbs. Table 1 lists the distribution of the part of speech tags in the larger lexicon.
Most of derived forms in common use are listed in the lexicon in their derived forms. Some irregularities in morpho-phonological process depend on the particular root word. Such irregularities are marked in the lexicon. More information on these irregularities and how they are dealt with in TRmorph is explained in the following subsections.

PoS	Count
Adjective	1244
Adverb	483
Conjunction	47
Interjection	131
Noun	23101
Postposition	36
Pronoun	21
Proper Noun	9532
Verb	2488

Table 1: Distribution of parts of speech in the lexicon.

	Low		High (I)	
	Rounded	Unrounded (A)	Rounded	Unrounded
Back	o	a	u	ı
Front	ö	e	ü	i

Table 2: Turkish vowels.

3.2. Morpho-phonological process

Turkish has a number of morpho-phonemic alternations that a morphological analyzer has to consider. These alternations are dependent on the phonological context, where the features of individual morphemes in the context affect this process. Before going through the morpho-phonemic alternations that are implemented in TRmorph, we will first review the alphabet of the language, the relevant features of the phonemes (or letters) and the analysis symbols used in this study.

3.2.1. Analysis and surface symbols

Even though we only deal with written text in this article, the Turkish orthography follows the sound patterns of the standard Turkish relatively loyally, and the same features can also be attributed to letters. Table 2 presents the Turkish vowels and relevant features, and Table 3 presents Turkish consonants classified into two classes based on voice feature.

Turkish alphabet has 8 vowels: a, e, ı, i, o, ö, u and ü. The most relevant grouping of the vowels are high vowels, and low-unrounded vowels as they play a role in vowel harmony and determine the surface realizations of a high number of morphemes. We denote these classes with capital letters I, A respectively in the descriptions below.[2] Since vowels in Turkish suffixes harmonize with the preceding vowel, the analysis symbol I will be realized as one of the high vowels, namely ı, i, u or ü, on the surface depending on the preceding vowel's rounded/unrounded and back/front features. Similarly, the analysis symbol A is realized as one of the low-unrounded vowels, namely a or e, depending on previous vowel's high/low feature.

Turkish alphabet has 21 consonants: b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y and z. Table 3 presents these consonants and relevant features. The most important feature that affects the morpho-phonological process is the voiced/voiceless difference. This feature plays a role in some of the phonological alternations described below. Table 3 shows voiced/voiceless consonants roughly sorted by their place of articulation—bilabial b/p (front) to glottal h (back). As in vowel classes, we present some groups of consonants formed by voiced/voiceless counterparts with capital letters. The last row of Table 3 lists the class labels that represent the consonants on the same column.

Besides the alternation of surface symbols, some analysis symbols may be deleted depending on the context. These are given in parentheses in the descriptions in the following subsections. For example, (n) represents a surface n that can be deleted and (A) represents one of the unrounded vowels that can be deleted in certain contexts.

3.2.2. Vowel harmony

Vowels in Turkish harmonize with the preceding vowel according to frontness and roundness.

- The analysis symbol A in a suffix is realized as e after a front vowel, and as a after a back vowel.

  For example, because of the vowel harmony, the plural suffix -l**A**r has two allomorphs -lar and -ler. Hence, the plural form of ev 'house' is ev-l**e**r, while plural form of oda 'room' is oda-l**a**r.[3]

- The analysis symbol I is realized as one of ı, i, u, ü, depending on roundness and frontness of the preceding vowel.

  For example, the evidential past tense marker -m**I**ş is realized differently for words gel, 'come', gör 'see', al 'take', and dur 'stop', resulting in surface forms gel-m**i**ş, gör-m**ü**ş, al-m**ı**ş and dur-m**u**ş.

One exception to these rules is also implemented, where A becomes I before the suffix -(I)yor. For example, gel-m**A**-(I)yor (come-NEG-CONT 's/he is not coming') is realized as gel-m**i**-yor. Similarly, ar**a**-(I)yor (call-CONT 's/he is calling') is realized as ar**ı**-yor.

3.2.3. Consonant voicing changes

Some consonants also change depending on the preceding or following context.

- Analysis symbols C and D at the beginning of suffixes are realized as their voiceless surface counterparts after voiceless consonants, and realized as their voiced surface counterparts after vowels.

  For example, the derivational suffix -CI becomes -**c**i after şeker 'sugar', but becomes -**ç**ı şarap 'wine', forming surface forms şeker-**c**i 'sugar maker/seller', şarap-**ç**ı 'wine maker/seller'.[4]

[2] In this paper we will use lowercase letters for surface alphabet, and use uppercase letters to represent analysis-only symbols. In the SFST implementation, special multi-characters labels are used for this purpose, hence, TRmorph can be used to analyze mixed-case documents.

[3] In these examples and in the examples that follow we use dash '-' in surface forms for clarity. This, of course, is not part of the real surface form of the words.

[4] Also note the different realization of the symbol I due to vowel harmony.

Voiced	b	m	v	d	z	n	l	r	c	j	y	g	ğ		
Voiceless	p		f	t	s				ç	ş		k			h
Analysis Symbol	(P)			(D)					(C)		(K)	(K)	(K)		

Table 3: Turkish consonants. The left-to right ordering roughly corresponds to the place of articulation from front (bilabial) to back (glottal). The last line of table presents analysis symbols used in TRmorph for these consonants in some morphophonological alternations.

- Analysis symbols C, D and P at the end of stems are realized as their voiced surface counterparts if followed by a vowel, and realized as their voiceless surface counterparts otherwise.

  For example the symbol P in lexical word kita**P** is realized as voiceless p if it is at the end of the word (kita**p** 'book') or followed by a consonant (kita**p**-çı, 'book seller') but as voiced b if followed by a vowel (kita**b**-ı 'book-ACC').

  The analysis symbol K also goes through a similar change, however, with a few exceptions. K is realized as

  - y only in suffix -mAK when followed by a vowel (gör-mA**K**-(y)I 'to see-ACC' is realized as gör-me**y**-i)
  - g only if preceded by n (ren**K**-(y)A 'color-DAT' is ren**g**-e)
  - in other contexts before a vowel, it is realized as ğ (kitap-CI**K**-(y)A 'book-DIMIN-DAT' is realized as kitap-çı**ğ**-a)
  - realized as k in all other contexts.

  As these alternations generally happen in some borrowed words with ending in voiced consonants b, c, d and g, it is common to assume that the lexical form ends with this voiced consonants, and devoiced on the surface if nothing or a consonant follows, otherwise stays unchanged. For this reason this process is commonly referred to as *final stop devoicing*. Since, the alternation is not predictable from the form of the root word, and some lexical items have to be marked as exceptions. In TRmorph, we chose to mark all lexical items which go through this alternation using special analysis symbols, and the rules described above take care of analyzing and generating the correct forms.

3.2.4. Buffer vowel or consonant drop

Some suffixes start with a consonant or vowel, that may be deleted depending on the context. These letters are commonly called *buffer* letters. The buffer letters are dropped if they are preceded by a letter from the same class, e.g. a buffer consonant is dropped if preceding letter is a consonant, but it is realized in the surface if preceding letter is a vowel.

For example, present continuous tense marker -(I)yor has a buffer I and locative case marker -(n)dA has a buffer n. The buffer is kept in gör-**ü**yor 's/he is seeing' and oda-**n**da 'room-LOC', but not in uyu-yor 's/he is sleeping' and ev-de 'house-LOC'.

3.2.5. Other alternations and exceptions

Besides above phonological alternations, there are a number of other alternations that are rare, or happen in exceptional cases.

- Some vowels in certain words are dropped if a suffix starting with a vowel is attached. For example bur**u**n 'nose' becomes burn-um (nose-P1S 'my nose').
- Some root final consonants are duplicated if followed by a suffix starting with a vowel. This generally happens with some borrowed words. For example hak-(n)I (right-P3S 'his/her right') is realized as hak**k**-ı
- The buffer (n) and (s) become y after words ending with su. For example, su-**(n)**I (water-P3S 'his/her water') is realized as su-**y**u.
- Some words ending in vowels borrowed from Arabic may cause the buffer s in the following suffix to be deleted. The most known example of this exception is that cami-(s)I 'mosque-P3S' is realized as cami-i. This seems to happen with a small number of borrowed words, and in TRmorph these words are marked in the lexicon. It appears that in current Turkish the form without s-drop, i.e. cami-si is even more common than the widely recognized s-drop form. Hence, we allow both surface forms in TRmorph. This is one of the few cases that TRmorph accepts multiple surface forms for the same analysis string.
- e in pronouns ben and sen becomes a if suffixed by the dative case marker. For example sen-(y)A (you-DAT 'to you') is realized as s**a**n-a.
- The passive morpheme alternates between In and Il depending on the last letter of the stem it is attached to. If the preceding letter is l the In for is used, otherwise Il for is used. For exmple, gör-ü**l**-di 'see-PASSV-PAST', but gel-i**n**-di 'come-PASSV-PAST'.
- A few words, including the pronouns o, bu, şu, get an additional n before suffixes. For example, o-DA 'he/she/it-LOC' is realized as o**n**-da

3.3. Morphotactics

Turkish words fall into two broad classes. Nouns, adjectives and adverbs form the *nominal* class. In Turkish, boundaries for these classes are rather blurred, and

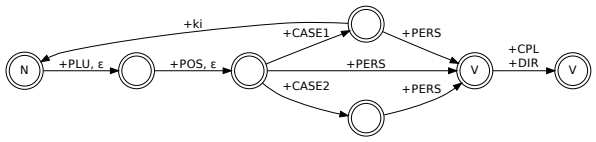

Figure 1: A simplified FSA for the nominal morphotactics.

they all share the same morphological properties. Nominal morphotactics is relatively regular and simple. Figure 1 presents a simplified finite state automaton describing nominal morphotactics. Verbs form the other class. As can be seen in Figure 2, the verbal morphotactics is more complicated and it also has more exceptions. It should be noted that these two figures are intended to give a general feeling of the 'beads on a string' morphotactics with only a few major divergences. The real picture is significantly more complicated, due to both irregularities and large number of allomorphs.

We will present nominal and verbal morphotactics separately here, however, the verbal and nominal morphotactics interact with each other. Both classes can receive suffixes that cause a possibly inflected word to change its class. For example, the verbal root in (3) takes a few verbal suffixes, then becomes a verbal noun by addition of a subordinating suffix and after a few nominal suffixes, and with the addition of a copular suffix it again functions as a verb.

(3) oku -ma -dık lar -ı -ydı
read -NEG -Part -PLU -POS -CPL
'They were the ones which s/he did not read.'

Most of the explanations below will cover only inflectional morphology. Except a few, most derivational affixes are rather unproductive. However, for some purposes, such as a fine-grained morphological segmentation, one may wish to analyze the lexicalized derivations as well. TRmorph distribution comes with a large inventory of derivational suffixes that can be used for similar purposes. By default, a number of productive suffixes are analyzed after the lexical forms. However, there are also a few cases, such as (4) below, where derivational suffixes may follow inflectional suffixes. TRmorph allows some of these productive derivational suffixes to be attached in certain points in the FSA given in Figure 1 and 2. The decision of using a derivational suffix by default, or allowing attachment to derived forms has been based on the coverage of the analyzer on the development corpus.

(4) oku -ma -mış lık
read -NEG -D_VN -D_NN
'The state of not being educated.'

Besides the morphotactics of the words described in this section, TRmorph also includes a specification for numbers.

3.3.1. Nominal morphotactics

The overview of nominal morphotactics in Figure 1 is relatively accurate. A root nominal can take the following suffixes, all the suffixes are optional, but when they co-occur they have to follow the order presented in the FSA.

- A nominal stem may be followed by *plural* suffix `lAr`.
- Next possible suffix that can attach to a nominal is one of 6 *possessive* suffixes. We label these suffixes in this paper as `-Pxy`, where 'x' one of `1`, `2` or `3`, which stands for first, second or third person respectively; and 'y' is either `S` for singular, or `P` for plural.
- Turkish has 5 commonly recognized *case*s: *accusative* (`-ACC:(y)I`), *dative* (`-DAT:(y)A`), *locative* (`-LOC:DA`), *genitive* (`-GEN:(n)In`) and *ablative* (`-ABL:DAn`). The suffix `-(y)lA`, sometimes regarded as *instrumental* (`-INS`) case, also behaves just like other case morphemes, and TRmorph also treats it as a case marker.

  Like other nominal suffixes, case suffixes are also relatively regular. However, the only notable divergence from the FSA in Figure 1 is due to different allomorphs—not only because of general phonological processes such as vowel harmony— of case morphemes following 3rd person possessive suffix and all other suffixes.
- If the word is in locative or genitive case, it can be followed by the suffix `ki`, after which all the nominal suffixes can again be added.
- All nominals can be followed by a verbal *person* agreement to form *nominal predicates*.
- A nominal predicate may optionally be followed by a *copula* or the *generalizing modality marker*, `DIr`.

The nominals include a large number of word classes, namely nouns, adjectives and adverbs. In TRmorph specification, only nouns are allowed to take the nominal suffixes presented in Figure 1. To allow adjectives and adverbs to also take the same suffixes, we allow all adjectival or adverbial stems to become a noun by a zero derivation.

3.3.2. Verbal morphotactics

The verbal morphotactics is more complex than nominal morphotactics. All edges in Figure 2 represent a large number of morphs, and there are quite few minor alternations, some of which has to be specified lexically.

The verbal roots can get a number of *voice* suffixes, the *negative* marker, a number of suffixes that form compound verbs. After these optional suffixes, a *person agreement* and a *tense/aspect/modality* marker are obligatory for finite verbs. A number of other optional suffixes, namely *copular* markers and the *generalizing modality marker* `-DIr` may follow. Alternatively, an 'untensed' may become a nominal by a number of subordinating suffixes which form *verbal nouns*, *participles* or *converbs*. The nominalized form may take some or all of the nominal inflections.

We will go through possible formation of verbs in more detail below. Due to space limitations, describing complete morphotactics would be impossible here. A more complete description is available in TRmorph distribution.

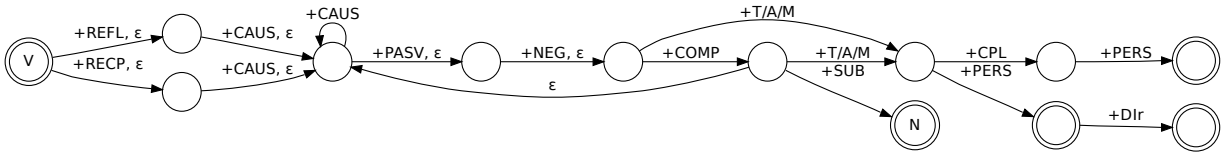

Figure 2: A simplified FSA for the verbal morphotactics.

- The *reflexive* and *reciprocal* suffixes are attached only to a small number of verbs, and unpredictable. The verbs that get reflexive, reciprocal, or both, are marked in the lexicon.

- When attached to root forms, causative morpheme shows quite some irregularity, taking one of six forms depending on the verb root. Otherwise it alternates between `-DIr` and `-t` depending on the preceding letter. The irregularities are handled by marking them in the lexicon. It is possible to repeat the causative suffix, as in `piş-`**`ir-t-tir`**`-di` 's/he arranged for it to be cooked'.

- The *passive* suffix takes two forms that are completely predictable from the preceding context, and handled by two-level rules.

- Negative marker precedes all other inflectional suffixes, except in one case explained in the following item. The *negative* marker `-mA` becomes `-mI` before the suffix `-(I)yor`. Like the alternation passive, this exception is dealt in morpho-phonological rules.

- The edge marked as `+COMP` represents 8 suffixes that form *compound verbs*, which generally express modality. Most productive of these are `-(y)Abil` and `-(y)Iver`. The others are not productive, however, TRmorph does not limit their use not to increase complexity, with the cost of over-generation and possibility of accepting some 'odd' constructions.

  The compound suffix `-(y)Abil` also has a form `-(y)A` which occurs before the negative marker and only in negative forms.

  The current version of verbal morphotactics specification in TRmorph has a lambda-transition after compound suffix, allowing attachment of causative suffix and suffixes that follow to a compound verb form. This handles relatively few constructions. However, it simplifies the FSA specification, with the expense of allowing some semantically odd constructions.

- After these optional suffixes, either the word may become nominal by receiving one of a number of *subordinating suffixes*, or become 'tensed' with a *tense/aspect/modality* marker.

  – Three main forms of subordination are possible. *Verbal nouns* can be formed with one of the 5 different suffixes, there are 3 suffixes that form *participles* and about 20 suffixes that form *converbs*. Some of these suffixes overlap, that is some suffixes, such as `-DIK` produce all three types of nominals. First two can combine all the nominal suffixes with some special cases. The converb markers are very selective with the suffixes they follow and precede. Current implementation of TRmorph does not restrict the combination of converbs with suffixes completely. In the applications that we have used TRmorph for, this did not cause serious problems. However, this creates a spurious ambiguity, and also causes overgeneration in generation mode.

  – There are 11 *tense/aspect/modality* markers that may appear in a verb, and together with a person agreement, use of one of these markers is obligatory to form a finite verb. The actual FSA implementing this part of the verbal morphotactics is rather complicated because of the preferences of each `T/A/M` suffix differ from each other both in respect to the preceding context and the person suffixes and the other optional suffixes that may follow. In addition, the *aorist* tense morpheme, shows quite some irregularity when it is attached directly to the root forms, which is unpredictable and has to be specified lexically.

- In a finite verb after `T/A/M` markers, there are two possible paths.

  – First, one of three copular markers which generally form complex tenses may follow. Generally after, but in some exceptional cases before the copular markers, a person agreement is compulsory.

  – Second, one of six person agreement morphemes followed by optional *generalizing modality marker*, `DIr`, may follow.

  The actual implementation of these in TRmorph is again complicated. The possible copular markers depend on the preceding `T/A/M` marker, and form of the person agreement depends on the preceding copula. Besides, the 3rd person plural agreement `-lAr` shows quite some variety with respect to at which point it attaches to a finite verb.

3.4. Limitations and future directions

- The main limitation of TRmorph is the incomplete and noisy lexicon. The rule set is fairly complete. However, there are a few rare cases that are not handled in

the morphotactics. Instead, a special lexicon that provides both desired analysis and surface strings is used for these exceptions.

- TRmorph currently only analyzes 'words'. However, there are some relatively productive morphological processes that go beyond white space boundaries. Reduplication of adjectives, restrictions of some morphemes only before certain particles (most notably to form converbs), and phonemic changes on some clitics because of preceding word are examples of such processes that may be implemented in later versions.
- Another problem arises because of a trade-off between complexity of the description and accuracy: some productive derivational morphemes are allowed by TRmorph to be used with any stem, which in fact may not be correct. This does not pose serious problems for analysis, however, it may be a problem for other applications.
- In general, TRmorph comes up with more analysis than one would initially expect. For example, every nominal in Turkish can take one of the person agreement suffixes to become a nominal predicate (e.g. `doktor-um` 'I'm a doctor'), and since 3rd person singular agreement is a null morpheme, every nominal is also analyzed as a nominal predicate. Methods of disambiguation exist for solving this problem (Hakkani-Tür et al., 2002).

4. Testing and Evaluation

The finite state transducers and the two-level specification are adequate and useful for building morphological analyzers. The development of such a system is very similar to programming: developers specify the task using a formal language and tools convert this information to a machine that does the task. Like the other commonly used tools for the same purpose, SFST provides a relatively easy to grasp formal language to specify the morphology. However, like many formal languages, the specification is not necessarily easy on unaccustomed eyes, and there are quite a few pitfalls that may haunt even the most experienced users. Particularly in finite state transducers, small finite state machines specified by the developer are combined into a single big FSA, and the interaction of the parts are not always so easy to anticipate. Therefore, like in any software development effort constant and reliable testing is important. As well as an error-free analyzer, a good morphological analyzer has to cover the language it is designed for well. The tests presented here aim to improve both the accuracy and the coverage of the analyzer.

With the hope that it may also be useful for others, this section first presents the methods used for testing TRmorph during its development. Afterwards, an evaluation of the system on two large corpora that has not been used during the development of the system is presented.

4.1. Test methods for two-level morphological analyzers

The obvious method to test a morphological analyzer during its development is to use relevant examples, such as the ones found in grammar books. A hand-crafted test set created alongside the rules specifying the morphology is indeed very useful, and TRmorph has also been tested with this method.

However, real world frequently presents examples that even most carefully crafted systems do not cover well. Testing the system on a large collection of words from the real-world data is the only way to discover some of the potential problems. During the development of TRmorph we used a large word list extracted from on-line newspapers. Since we do not have a gold-standard analysis of the words, finding problems by analyzing such a list is non-trivial. In this study, we have used an unannotated list of words and paid attention to the following:

- Changes in analysis of word list after changes to the rule set. Even though the complete list is difficult to manually check, the differences in analysis after a small change in the specification are generally easy to inspect. The unexpected changes are a good indication of problems.
- The words that produce high number of analyses are also found to be good indication of problems with the analyzers.
- High frequency words with no analysis also tend to indicate problems with the analyzer. This tends to find problems with the specification of the morphology in the early stages of the development. Later on, it is mostly useful in spotting frequent out of vocabulary words.
- Mismatches between the original input and the word list obtained using an analyze-generate. Since finite state transducers are generators as well as analyzers, one can feed the analysis in generation mode, and obtain a list of surface strings. If the morphological specification includes analysis strings with multiple surface realizations, there will be a natural mismatch. However, after filtering out the expected differences, the remaining discrepancies are useful for spotting errors in the analyzer. Particularly, generated words that never occur in input corpora, are good indications of the problems with the specification.

4.2. Evaluation

The quality of a morphological analyzer can be measured based on how well it analyzes the real language data. TRmorph has been constantly tested on real world data during its development. However, to eliminate the possibility of a bias introduced during the development, we present a few evaluation metrics on two corpora that have not been used in development of the system before.

For this purpose, TRmorph has been tested on two relatively large corpora: the METU corpus (Say et al., 2002) and Turkish Wikipedia (as of 2009-10-16). Both corpora have been preprocessed by removing tags, punctuation and numbers. The number of word types and tokens in each corpus, along with the percentage of the analyzed words are given in Table 4. To give a picture of the morphological

Corpus	Number of Words		Analyzed (%)	
	Type	Token	Type	Token
METU	162724	1431513	88%	95%
Wikipedia	1120779	29143186	34%	85%

Table 4: Coverage on METU and Wikipedia corpora.

Error type	METU		Wikipedia	
	Top	Rnd	Top	Rnd
Proper names	0	17	2	8
Abbreviations/Terms	63	10	21	6
Other OoW words	2	16	0	6
Foreign words	15	42	53	20
Typo/alt. spelling	20	15	24	60

Table 5: Error analysis for 100 most-frequent (Top) and 100 random (Rnd) unanalyzed words.

complexity of the real-world data: TRmorph found 2.77 morphemes per word type on average and the number of words that received an analysis with 5 or more surface morphemes consists of 10% of the word types. Both numbers are lower on Wikipedia word list, the mean number of morphemes are 1.14 and only 6% of the words received analyses with 5 or more morphemes. This seems to be mainly due to large number of proper nouns in Wikipedia corpus.
The low coverage on Wikipedia corpus seems to be due to the large number of foreign words[5], large number of proper nouns that are not in the lexicon, and a surprisingly high level of typos. To give a quantitative indication of the unanalyzed words, we picked two sets of unanalyzed words, 100 most-frequent, and 100 random. For these sets of words, we have manually identified the reason for the failure. Table 5 presents these words broken down to 5 categories. All unanalyzed words are out of vocabulary words, mostly domain specific terms or proper names.
The same process is repeated for successfully analyzed words, where there was no errors, but a some (spuriously) ambiguous analyses.

5. Summary and Conclusions

In this paper we presented TRmorph, a two-level morphological analyzer for Turkish. The accuracy and coverage of TRmorph tested on real-world data met our expectations. Except for a few known cases listed in Section 3.4. the analyzer does not make systematic mistakes on real-world large corpora. Most errors are due to out of vocabulary words.
As stressed throughout the paper, the strength of TRmorph lies in its availability. Even though it is not the first one— and possibly not the best either— TRmorph has been implemented with free tools and resources, and distributed with a license that allows free use, modification and distribution. Consequently, it can be used and modified freely for different tasks, and may form a base for creating morphological analyzers for related languages. We also hope to improve it further by corrections and comments from a larger user base.

[5] 'the' is the 21st most frequent word in this corpus.

6. References

Ahmet Afşin Akın and Mehmet Dündar Akın. 2007. Zemberek, an open source nlp framework for Turkic languages. Available at http://zemberek.googlecode.com/.

Çağrı Çöltekin and Cem Bozsahin. 2007. Syllables, morphemes and Bayesian computational models of acquiring a word grammar. In *Proceedings of 29th Annual Meeting of Cognitive Science Society*, pages 887–892, Nashville.

Aslı Göksel and Celia Kerslake. 2005. *Turkish: A Comprehensive Grammar*. London: Routledge.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410.

Jorge Hankamer. 1986. Finite state morphology and left to right phonology. In *Proceedings of the West Coast Conference on Formal Linguistics*, volume 5. Stanford Linguistic Association.

Arne Fitschen Helmut Schmid and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition, and inflection. In *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1263–1266.

Lauri Karttunen and Kenneth R. Beesley. 2005. Twenty-five years of finite-state morphology. In Antti Arppe, Lauri Carlson, Krister Lindén, Jussi Piitulainen, Mickael Suominen, Martti Vainio, Hanna Westerlund, and Anssi Yli-Jyrä, editors, *In Inquiries into Words, Constraints and Contexts: Festschrift for Kimmo Koskenniemi on his 60th Birthday*, CSLI Studies in Computational Linguistics Online, pages 71–83. CSLI Publications, Stanford, California.

Jaklin Kornfilt. 1997. *Turkish*. London and New York: Routledge.

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and generation*. Ph.D. thesis, University of Helsinki.

Geoffrey Lewis. 2000. *Turkish Grammar*. Oxford University Press, second edition.

Brian MacWhinney and Catherine Snow. 1990. The child language data exchange system: an update. *Journal of Child Language*, 17(02):457–472.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9.

Tommi Pirinen. 2008. Automatic finite state morphological analysis of Finnish language using open source resources. Master's thesis, University of Helsinki.

Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut Özge. 2002. Development of a corpus and a treebank for present-day written Turkish. In *Proceedings of the Eleventh International Conference of Turkish Linguistics*.

Helmut Schmid. 2005. A programming language for finite state transducers. In *Proceedings of the 5th International Workshop on Finite State Methods in Natural Language Processing (FSMNLP 2005)*, pages 308–309.

Eros Zanchetta and Marco Baroni. 2005. Morph-it! a free corpus-based morphological resource for the Italian language. *Corpus Linguistics 2005*, 1(1).