Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 3837 – 3851
© TÜBİTAK
doi:10.3906/elk-1902-125

TRMOR: a finite-state-based morphological analyzer for Turkish

Ayla KAYABAŞ[1,*], **Helmut SCHMID**[2], **Ahmet E. TOPCU**[3,1], **Özkan KILIÇ**[1]
[1]Department of Computer Engineering, Faculty of Engineering, Ankara Yıldırım Beyazıt University, Ankara, Turkey
[2]Center for Information and Language Processing, Ludwig Maximilian University, Munich, Germany
[3]College of Engineering and Technology, American University of the Middle East, Eqaila, Kuwait

Received: 18.02.2019 • **Accepted/Published Online:** 24.05.2019 • **Final Version:** 18.09.2019

Abstract: Morphological analysis is an important component of natural language processing systems like spelling correction tools, parsers, machine translation systems, and dictionary tools. In this paper, we present TRMOR, a morphological analyzer for Turkish, which uses the SFST tool (Stuttgart Finite-State Transducer). TRMOR can be freely used for academic research (see http://www.cis.uni-muenchen.de/ schmid/tools/SFST/). It covers a large part of Turkish morphology including inflection, derivation, and some compounding. It uses morphotactic and morphophonological rules and a stem lexicon. We describe the morphological structure of Turkish, explain the phonological and morphological rules implemented in TRMOR, evaluate the system, and test it in special cases. The evaluation of TRMOR was executed on gold-standard words. One thousand words were randomly selected from Wikipedia word lists. For those words, we achieved gold-standard analysis. TRMOR has 94.12% precision on these 1000 words that were randomly selected from Wikipedia word lists. Morphological analyses of Turkish are prepared for the gold-standard version since, to our knowledge, there is no gold-standard segmentation available for Turkish morphological analyzers for noncommercial purposes.

Key words: Finite-state morphology, Turkish morphology, gold standard

1. Introduction

Finite-state systems and their properties are based on mathematical objects. Therefore, they can be implemented well, and finite-state transducers (FSTs) can be applied efficiently to natural languages. A FST is a graph consisting of a set of nodes (called states) and edges (called transitions). The set of states comprises a single start state and a subset of final states. The transitions are directed and labeled with symbols (e.g., letters). FSTs are said to accept an input string (or word) if a path exists from the start state to the final state, such that the concatenation of the symbols on the transitions of the path forms the input string. For example, the FST in Figure 1 accepts the strings *"ola"* and *"olala"*, etc.

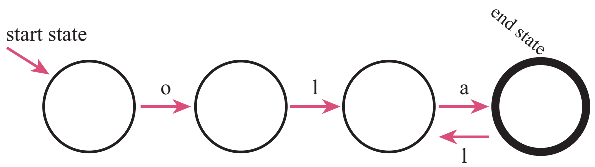

Figure 1. A simple FST.

[*]Correspondence: ayla.kayabas0@gmail.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

An example of the application of FSTs is morphological analysis, which splits an inflected word into its morphological information and returns it [1]. The finite-state mechanism is the basis for morphological analysis. Morphological analysis gives the grammatical and derivational properties of a word. In the design of finite-state morphology, one should plan and implement mechanisms that carry out the phonological alternatives of the respective language and check the validity of the word for realization.

Morphological analysis is the decomposition of a word into its smallest components, called morphemes. For strongly inflected languages such as Turkish, Finnish, or Hungarian, morphological analysis is very important since it identifies and categorizes the stem of the word and its class and determines its affixes. Almost all morpheme types, which are individual words in nonagglutinative languages, are included/represented in one word (in a prescribed order). This concatenation of the prepositions and pronouns into one large string leads to data sparseness. The large number of suffixes that may occur in different orders leads to a large number of possible word forms (sparse-data problems). Suffixes therefore play an important role by encoding much more grammatical information than in a configurational language, such as English. The agglutinative nature of the morphology is responsible for poor performance of the finite-state based morphological analyzers [2] for languages, such as Turkish. The affixes play an important role in the structure of the language. The majority of the stems are bound and determine the class of the words in the language. The large number of word forms produce sparse-data problems, which are solved by decomposing a word into smaller, more frequent units. The reduced analysis of word forms also helps to solve sparse-data problems, statistically extracting from many word forms with only the relevant features [3]. In other words, NLP techniques, such as FST, are crucial to solve the data sparseness problem with some languages, especially agglutinating languages such as Turkish.

The most important word formation process in Turkish is suffixation, which forms a new word by attaching a suffix morpheme to the right side of the string. Most Turkish suffix morphemes have more than one [4] surface realization (allomorphs). Even one morpheme can represent up to 16 different forms due to vowel and consonant harmony. Therefore, one morpheme can have different surface forms [5]. The canonical form of perfect morphemes is illustrated in Table 1. The initial consonant of some suffixes and the vowels of almost all suffixes depend on the preceding consonants or vowels. The plural morpheme, for example, has two allomorphs: -lar as in *kitaplar* (books) and -ler as in *kediler* (cats). Because of the vowel harmony, the vowel of the morpheme is realized as either /a/ or /e/ depending on the backness and roundedness of the most recent vowel. The perfect morpheme for example has 8 forms since both the consonant and the vowel undergo variation.

Table 1. Perfect suffix alternation.

dı/tı	di/ti	du/tu	dü/tü
kal-dı (s/he stayed)	gel-di (s/he came)	oku-du (s/he read)	gül-dü (s/he laughed)
kaç-tı (s/he flew)	git-ti (s/he went)	koş-tu (s/he jogged)	küs-tü (s/he was angry with)

In this paper, we implement a Turkish morphological analyzer called TRMOR, using the Stuttgart Finite-State Tool (SFST) [1], both of which are freely available under the GNU public license. TRMOR is a two-level morphology, realized as a FST. The two-level morphology assures that the direction of analysis is reversible. The SFST provides a programming language based on extended regular expressions, which are compiled into FSTs. TRMOR is also a rule-based morphological analyzer, like other morphological analyzers that use SFST

[6]. The implementation components of TRMOR consist of a lexicon in which the stem morphemes are listed with their part-of-speech and inflection class, a set of regular expressions specifying morphotactics, and a set of phonological rules transforming the canonical representation of the morphemes to their surface forms. Because of the highly productive nature of agglutinative languages, the set of possible word forms cannot be listed as in the case of English; rather, it needs to be analyzed and tested. We evaluated TRMOR using a Turkish corpus with gold-standard morphological annotations and compared it to other morphological analyzers.

2. Related works

Morphological analyzers are the base components of applications to recognize words. Turkish morphology was studied with different formalisms of finite-state techniques. The work of Jorge Hankamer [7] described a finite-state morphology and left-to-right phonology. The difference between Hankamer's work and TRMOR regarding vowel harmony will be explained in detail in Section 6.

The well-known application of finite state systems in morphology is two-level morphology. Two-level morphology is based on finite-state mechanisms. Some tools of this mechanisms, such the PC-KIMMO system developed by Antworth [8], the XFST Xerox finite state tool (which is commercial) by Karttunen [9], foma: a finite-state machine toolkit and library by Huldén [10], and Helsinki Finite-State Transducer Technology (HFST), have been implemented using the SFST and OpenFst software libraries [11], among others. Some morphological analyzers have been developed for Turkish as well. The renowned ones were developed by Oflazer [12]. The first finite-state-based mechanism for Turkish was implemented by Hankamer [13], although neither one of these is freely accessible.

The main forms of finite-state mechanisms are phonology and morphology [14]. The state-of-the-art application of this representation is the SFST system. The system is also used by analyzers for the Turkish language, namely TRMOR and TRmorph [15], the 2010 version of which was used in this study. The 2013 version of TRmorph is a complete overwrite of the previous version. TRmorph is the first freely available finite-state-based morphological analyzer for Turkish, which in contrast to TRMOR covers almost all morpheme types, inflections, derivations, and compounding. SFST is used to build morphological analyzers for a number of other languages differing in morphological complexity. Morphological analyzers developed using SFST include a finite-state morphology for German, SMOR [6], and LATMOR for Latin [16], while for English morphology EMOR has been developed by Helmut Schmid using the SFST finite-state transducer toolkit [1] and the morphological resources of Karp et al. [17].

3. The lexicon

A lexicon is an additional file in the SFST. It is created manually by inputting words for checking. The TRMOR lexicon contains 36,903 entries. Each entry consists of a morpheme with information about the morpheme type, the part-of-speech, and the inflection class. The distribution of word classes in TRMOR is presented in Table 2. The lexicon entries in TRMOR encode features of words (see Table 3). The system includes rules, which derive derivation and compounding stems from base stems [6]. Each base stem belongs to a word class and an inflectional class, which produces the inflectional endings for each case, number, gender (note that there is no grammatical gender in Turkish and therefore no gender encoding in TRMOR), and person variation of the lemma [18].

For rule-based word formation, the inflection class plays an important role. The nouns in the lexicon have 4 inflection classes, namely ‹NomReg-p›, ‹NomReg›, ‹NomReg-su› and ‹NomRegCop›. ‹NomReg-p› occurs

Table 2. Distribution of word classes.

Word class	Total
N	23,214
V	2638
ADJ	1517
NE	9534

Table 3. Lexicon entries of TRMOR.

Types	Tags
Entry types	‹Stem›‹Suffix›
Stem types	‹base›‹deriv›‹compound›
Word types	‹V›‹ADJ›‹N›‹NE›
Inflectional classes	‹NomReg›‹VerbReg›...

16,419 times more than ‹NomReg› at 6751. As we noted above, the ‹NomReg-p› inflection class comprises nouns that end with consonants. The inflection classes comprise the verbal and nominal categories (nouns, adjectives, adverbs). These differ in their root words' endings again. Some nominal lexicon entries with their morpheme types and inflection classes are listed in Table 4. In order to obtain an inflected word form, we apply an inflectional suffix to a base stem, such as oku+yor → okuyor.

Table 4. Example entries from the lexicon with their glosses.

Entry	Lemma	Gloss	Part of speech	Type	Origin	Inflection class
Stem	silgi	eraser	N	base	native	NomReg
Stem	kalem	pencil	N	base	native	NomReg-p
Stem	su	water	N	base	native	NomReg-su
Stem	ev	house	N	base	native	NomRegCop
Stem	Mehmet	Mehmet	NE	base	native	NEReg

For the first lexicon entry, the word *silgi* ('eraser') is analyzed. Its word class is assigned as noun (‹N›), its stem type as base stem, its origin as native, and its inflection class as ‹NomReg›. ‹NomReg› is used as a trigger, which at the same time assigns the variable $NomReg$. The second and fourth expressions are for the lexicon entries *kalem* ('pencil') and *ev* ('house'), with their lemmas, their part-of-speech tags (in these examples ‹N› for nouns), their origin (native), and their inflection class (‹NomReg-p›). For the fourth lexicon entry defined with *ev*, which belongs to inflection class ‹NomRegCop›, all nouns can be analyzed as copulas. We distinguish between the nouns that end with vowels and consonants in their inflection class. As one can see from lexicon entries *kalem* and *silgi*, the second lexicon entry belongs to a different inflection class. The third lexicon entry, *su* ('water'), differs from other nouns in declination and case, and since there are many words ending in *su*, we treat it as a separate lexicon entry. Therefore, this word has a distinct inflection class. The fifth lexicon entry is a personal name and belongs to the inflection class of named entities.

The derivational suffixes define the features of a derived word form, namely the word class, the inflectional class, and the origin. These features follow the suffix morpheme in the lexicon entry of the suffix [6]. The concept of derivation is as usual in the following way:

‹stem›word ‹word class›‹stem type›‹origin›
‹affixtyp›‹simplex›‹wordclass›‹stem type›‹origin›‹suffix›‹word class›‹stem type›‹origin›‹flexion class›

The representation of noun to verb (1) and noun-to-noun (2) derivations in TRMOR are shown as in the lexicon:
(1) güzel+leş → güzelleş ($N \rightarrow V$)
‹stem›güzel ‹N›‹deriv›‹native›
‹suffix›‹simplex›‹N›‹deriv›‹native›l‹K›ş ‹V›‹base›‹native›‹VerbReg-Consonant›

(2) güzel+lik → güzellik $(N \to N)$
‹stem›güzel ‹N›‹deriv›‹native›
‹suffix›‹simplex›‹N›‹deriv›‹native›l‹Y›k‹N›‹base›‹native›‹NomReg-p›

4. TRMOR: A finite-state-based morphological analyzer for Turkish

In the design of finite-state morphology, one should design and implement mechanisms that carry out the phonological alternatives of the particular language and check the validity of words for their realization.

A regular relation corresponds to a FST [19]. Implementing a morphology in SFST means incrementally building a complex regular expression, which is then compiled into a transducer. TRMOR first concatenates stem and suffix morphemes in all possible correct sequences via morphotactic rules, and then maps the resulting string to the correct surface form via morphophonological rules [6].

The surface level realizes the phonological form of a string through a chain of symbols in the normal inflected form of a word. The lexical level fills the abstract basic form of input with morphotactic and lexical information, or individual morphemes of a word (from entries in a lexicon). Apart from very few exceptional cases, the surface realizations of the morphemes in Turkish are conditioned through various regular morphophonological processes like vowel harmony and consonant assimilation and deletion. Table 5 shows some surface forms in examples. Here F stands for morpheme margin, and K and Y denote triggers for vowels.

Table 5. Surface forms in examples.

Surface level	String	Analysis level
atın	at ('horse')	at‹N›‹F›‹Y›n
çiçekte	çiçek ('flower')	çiçek‹N›‹F›‹t›‹K›
okuyorsunuz	oku ('read')	oku‹V›‹F›‹Y›yor ‹F›s ‹Y›n ‹Y›z

There are two major phenomena that morphophonological rules need to account for: vowel and consonant harmony. These influence most Turkish word formations. The lexicon and the morphotactic and morphophonological rules will be discussed in the next sections. The morphotactic word forms can be quite complex when multiple derivatives are involved [20]. The morphotactical order for verbal suffixes that we use in TRMOR is given in Figure 2.

4.1. Morphotactic process in TRMOR

Derivational suffixes change the part of speech while inflectional suffixes represent features such as number and case. In TRMOR, word formation happens through the concatenation of bound morphemes with other bound morphemes, such as derivational or inflectional morphemes. Free morphemes such as -mı/mi build yes/no questions [4]. The entry types consist of stems and suffixes. The stem types of TRMOR are composed of base stems, with the derivation participle, or as in the example *yapıyor idiysen* ('If (as you imply) you did'). However, this suffix –idiysen, separated into morphemes as idi (=perfect of to be) + yse (cond. causal) + n (2sg), usually attaches to tense suffixes, e.g., -Yyor, -mYş, -dY, and becomes *yapıyorduysan* ('If (as you imply) you were doing') in this case. An interesting example of word formation in Turkish is given in Table 6 with the following example:

ölümsüzleştirilemeyeceklerdenmişim (I am not one of those who never converted to be immortalized).

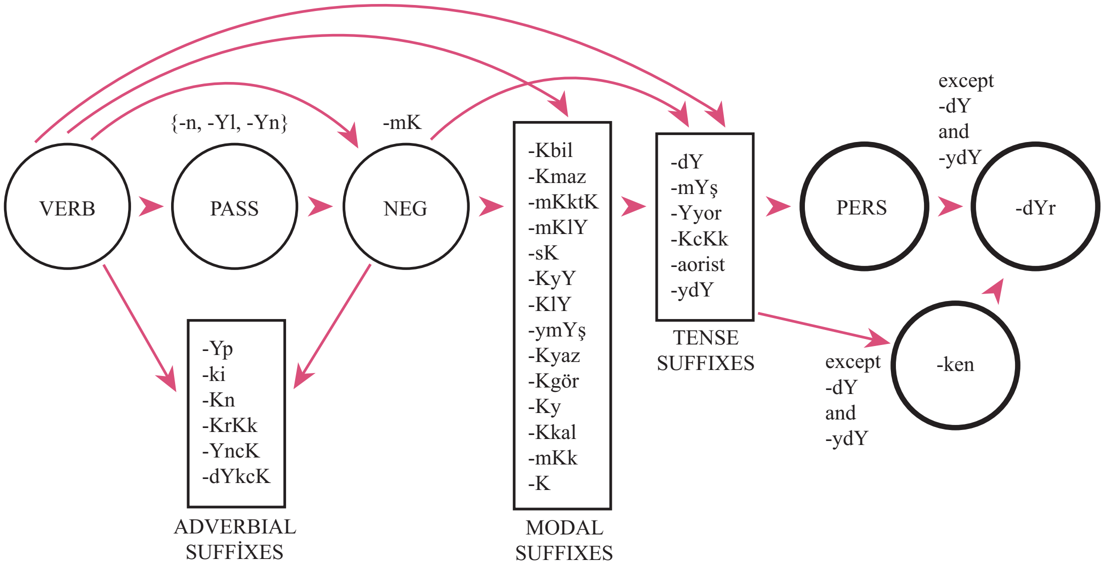

Figure 2. Illustration of morphotactic orders for verbs used in TRMOR.

Table 6. An illustration of Turkish suffixation.

root	1	2	3	4	5	6	7	8	9	10	11	12	13
öl	üm	süz	leş	tir	il	e	me	y	ecek	ler	den	miş	im
die	D_N	D_ADJ	D_V	caus	pass	opt	neg	inter-fix	fut	pl	abl	past	1.sg

This extremely complex example, concatenated with 12 suffixes (except the interfix y in position 8), is a prototypical illustration of suffixation from derivational level (see Table 7 for some derivational suffixes implemented in TRMOR) to the inflectional level in Turkish. This type of suffixation may be encountered in daily use of language. Here, from the verb *öl* ('die'), with the first derivational suffix -Ym (here -üm because of vowel harmony, as addressed in Section 6.1), it becomes a noun; with the second derivational suffix –sYz (=without) (here Y becomes /ü/), the first derived word, *ölüm* ('dead'), is converted into an adjective, *ölümsüz* ('immortal'), and then from the adjective, with the third derivational suffix -lKş, it again becomes a verb, *ölümsüzleş* ('become immortal').

In Table 7, some examples of derivational suffixes implemented in TRMOR are given with their derived directions, examples, and glosses.

Table 7. Some derivational suffixes with examples and their glosses.

Derivational suffix	Derivation direction	Example Turkish	Gloss
-dYk	V → ADJ	gittiğim	that I had gone
-YcY	N → N	kalıcı	staying
-cK	N → N	Fince	Finnish
-cY	N → N	Türkçeci	Turkish teacher

TRMOR recognizes the productive and a few unproductive derivational suffixes. Dealing with all unproductive suffixes would have been too much work. Most of the Turkish derivations are usually formed by suffixation. Some derivational suffixes are inflectional, too, or vice versa. For example, the -Yn suffix can occur as a passive suffix, such as in the word *kalınır* ('one stays'), and as a derivational suffix, such as in the word *basın*, from *basmak* ('publish'), *bas+Yn* (1press'). We defined this suffix once as derivation in the lexicon and implemented it for a passive suffix as a phonological rule. It is defined only as inflectional passive suffix +Yn. For the word basın ('press'), there is a lexicon entry for the derivational suffix +Yn to prevent oversegmentation. There are different morphemes with the same surface forms in Turkish. We treated such morphemes differently in the lexicon as explained in the implementation section. For semantic reasons, most suffixes in Turkish, especially the derivational ones, can be attached to only certain stems [21]. The inflection classes of suffixes differ also according to their ending in TRMOR. The vowel-ending suffixes belong to the $VerbReg - p$ inflection subclass and the consonant ending suffixes to the $VerbReg - Ym$ inflectional subclass. These subclasses define the personal pronouns. We will briefly look at the inflection classes for the lexicon entry *belirle* ('determine') in Figure 3:

‹Stem›belirle‹V›‹base›‹native›‹VerbReg›

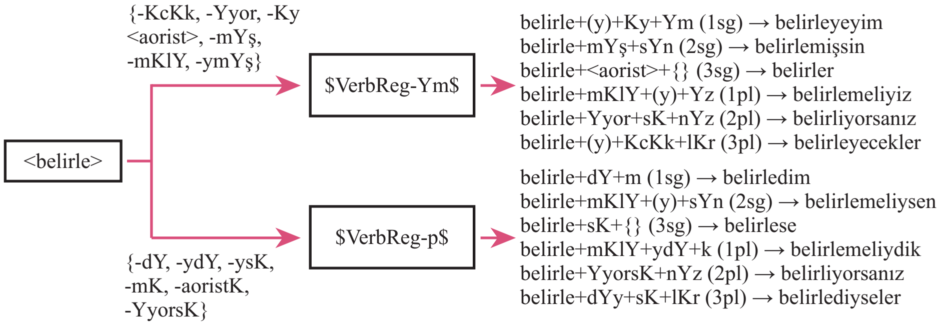
{-KcKk, -Yyor, -Ky
<aorist>, -mYş,
-mKlY, -ymYş}
<belirle>
$VerbReg-Ym$
$VerbReg-p$
{-dY, -ydY, -ysK,
-mK, -aoristK,
-YyorsK}
belirle+(y)+Ky+Ym (1sg) → belirleyeyim
belirle+mYş+sYn (2sg) → belirlemişsin
belirle+<aorist>+{} (3sg) → belirler
belirle+mKlY+(y)+Yz (1pl) → belirlemeliyiz
belirle+Yyor+sK+nYz (2pl) → belirliyorsanız
belirle+(y)+KcKk+lKr (3pl) → belirleyecekler
belirle+dY+m (1sg) → belirledim
belirle+mKlY+(y)+sYn (2sg) → belirlemeliysen
belirle+sK+{} (3sg) → belirlese
belirle+mKlY+ydY+k (1pl) → belirlemeliydik
belirle+YyorsK+nYz (2pl) → belirliyorsanız
belirle+dYy+sK+lKr (3pl) → belirlediyseler

Figure 3. Visualization of the inflectional variants of personal suffixes after some suffixes are implemented in TRMOR using the inflection classes ‹VerbReg-p> and ‹VerbReg-Ym> using the verb *belirle* (= "determine").

We list all possible inflectional surface variants for *belirle* based on the inflection class ‹VerbReg›. The variable ‹VerbReg› can go into two suffix inflection classes. The main inflection class can be used without damaging the morphotactical order first divided into two inflection classes because of vowel harmony. Each class contains information about inflectional endings, which determine the inflection class of personal suffixes regarding the preceding suffix. Visualizations of the inflectional variants of personal suffixes are given after some suffixes are implemented in TRMOR using the inflection classes ‹VerbReg-p› and ‹VerbReg-Ym› using the verb *belirle* (= 'determine').

4.2. Morphophonological process in TRMOR

There are two styles [14] of implementing morphophonological rules, parallel and sequential, upon which the TRMOR transducer is built. The rules of our system are as follows. When a compiler reads the lexicon file, each line is converted to a transducer, and then transducers are combined with the OR-operator. The following presentation executes the rules sequentially one at a time.

"lexicon" || R1 || R2 || R3 || R4 || R5

Dictionaries and rules are compiled together into a finite-state transducer. Different types of linguistic characteristics can be realized by the FST [23]. In sequential systems, the surface form of the lexical form is derived using ordered rules through a series of intervening representations. In a parallel description, the rules directly restrict the execution of the lexical form without access to an intermediate phase. Although it could be the case that a parallel rule takes precedence over another, it does not mean that their application is temporarily ordered.

In 2-level morphology all morphophonological rules are applied in parallel with identical input and output strings and produce the surface string in one step. The parallel rules may interact with each other in complicated ways. The other style applies the rules in sequence [14]. Each rule receives the output of the previous rule as an input and sends its output to the next rule. The sequence of rules generates multiple intermediate representations before producing the final surface form. The compiler later collapses the sequence of processing steps into a single step, which directly generates the surface form from the analysis string (in generation mode) or vice versa (in analysis mode). TRMOR uses 44 rules to map morpheme sequences to surface forms. The inflectional and derivational suffixes contain the special trigger symbols Y and K (instead of vowels), which undergo harmonization. Table 8 and Table 9 show to which vowels these trigger symbols are converted, depending on the preceding vowel, which forms the context.

During the design and implementation of the morphophonological rules, it is important to choose the correct order of the rules in order to avoid incorrect analyses. We tried to make the rules as general as possible. Usually there are separate rules for morphemes ending in a vowel and for morphemes ending in a consonant.

Table 8. The vowel harmony rule for Y.

Trigger/context	aı	ei	ou	öü
Y	ı	i	u	ü

Table 9. The vowel harmony rule for K.

Trigger/context	aıuo	eiöü
K	a	e

4.2.1. Vowel harmony

Vowel harmony, a characteristic feature of Turkish morphology, states that all vowels must match within a domain (usually the noncompound word) in terms of one or more properties [24]. Figure 4 shows how the harmony rules transform the trigger symbols ‹K› and ‹Y› to the correct vowels depending on the previous vowel [25]. The harmony rules are implemented in TRMOR by first mapping each trigger symbol to all of its possible realizations and then filtering out vowel sequences that violate the vowel harmony constraints. The resulting word form is *koşuyorduysanız* (‘if (as you imply) you were running’) and the analysis level is as follows:

koş‹V›‹praes›‹di_past›‹cond_cop›‹2›‹pl›

A special phenomenon occurs with the Turkish present tense morpheme -iyor/-ıyor in the case of polysyllabic verb roots that end in a vowel. The final stem vowel here is deleted before the vowel harmony rules are

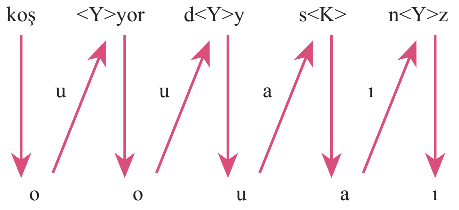

Figure 4. Stepwise vowel harmony.

applied. For the verb *oyna* (‘play’), for example, we obtain the following:

oyna + ⟨Y⟩yor + ⟨Y⟩
oyn + ⟨Y⟩yor + ⟨Y⟩
oyn + uyor + ⟨Y⟩m
oyn + uyor + um
oynuyorum

The preceding rounded back vowel /o/ triggers the realization of Y as /u/. If we applied vowel harmony first, the preceding ‘a’ would instead trigger the incorrect realization of Y as /ı/. The following example shows a different approach to vowel harmony, as proposed by Hankamer [7].

Stem → öde + Yyor
Vowel deletion of suffix → öde + yor
Vowel raising → ödi + yor
Rounding → ödü + yor
Surface form → ödüyor

Hankamer first deletes the initial vowel of the suffix -Yyor and then applies vowel harmony rules to the last vowel of the stem, which is raised and rounded.

öde+yKcKk ⇒ ödi+yecek ⇒ ∗ ödüyecek
söyle+yKcKk ⇒ söyli+yecek ⇒ *söylüyecek

In contrast, our rules produce the correct results:

öde + KcKk ⇒ öde + (y)+ ecek ⇒ ödeyecek

For the suffix KcKk, no vowel is deleted. The vowel harmony rules replace the trigger symbol K with /e/ and finally the joint element ‘y’ is inserted. It seems that our harmony rule approach is simpler and more reliable than Hankamer’s, as the example shows [7]. His approach is simple only for the two verbs *de* (‘say’) and *ye* (1eat’), handled as exceptions and defined as a rule in TRMOR. These verbs are highly irregular in their inflection and require special treatment. When the present tense morpheme -Yyor is added, the single vowel in the stem is deleted after applying the vowel harmony rule to Y, resulting in the form. In the case of the suffix -KcKk, a joint element /y/ is inserted after applying the vowel harmony rule for K, resulting in the form.

The vowel harmony rules for polysyllabic verbs first check which types of vowels occur in the root. If there is a low and a high vowel (see Table 8), then the last vowel is deleted and the -Yyor (present suffix) is chosen according to the remaining high vowel. For example, the verb *zıpla* (‘jump’) has a high vowel /ı/ and a low vowel /a/. /a/ is deleted, and Y is realized as the high vowel /ı/ because of the remaining vowel /ı/. Zıpla+Yyor becomes *zıplıyor* (‘he/she is jumping’). Y represents the high vowels. When there exist two vowels (in polysyllabic cases, one considers only the first occurring vowels), there is a matter of low vowels (see above in Table 8) and Y takes the high vowel of the rounded vowel.

4.2.2. Consonant harmony

If the stem ends in a voiced consonant and the initial consonant of the suffix is an alternating consonant (d/t, c/ç, g/k), then the initial consonant of the suffix becomes voiced (d, c, g); otherwise, it is unvoiced. In the word form *kırda* (‘on the field’), for example, the last stem phoneme /r/ is voiced and the suffix begins with an alternating consonant. According to the rule, the voiced variant of the ablative suffix -da is realized.

4.2.3. Final devoicing

The voiced consonants /b/, /d/, and /g/ become voiceless at the end of the word: for example, the singular form of the noun stem *kitap* (‘book’) with accusative suffix (-Y) kitap+Y *kitabı*. However, this does not always work with monosyllabic words: for example, the word *hap* (‘tablet’). Namely, hap+ı remains as *’hapı’*, not *habı*, like *haç* (‘crucifix’) to *haçı*, *saç* (‘hair’) to *saçı*, *kat* (‘floor’) to *katı*, etc.

4.2.4. Dropping vowels

Some Turkish words drop the last vowel when a suffix starting with a vowel is added. The accusative form of the word stem *akıl* (’intelligence’), for instance, is *aklı*. Whether the vowels are dropped depends on the stem and is not predictable. In the morpheme lexicon, we therefore add a trigger symbol at the point where the deletion will take place and define a phonological deletion rule, which is triggered when a suffix that begins with a vowel is added.

4.2.5. Dropping the plural suffix

The plural suffix -l‹K›r (3.pl.) is completely dropped if it follows a possessive suffix -l‹K›r‹Y›.

4.2.6. Consonant duplication

The final consonant of a morpheme is duplicated if it is directly preceded by a vowel. For example, the word *hak* (‘right’) becomes *hakkı* when the accusative suffix -Y is added, while the trigger ‘X’ is assigned for the duplication. In Figure 5, one can see other constraints of the word *hak*, namely in the forms of plural and singular, again in accusative, dative, and genitive.

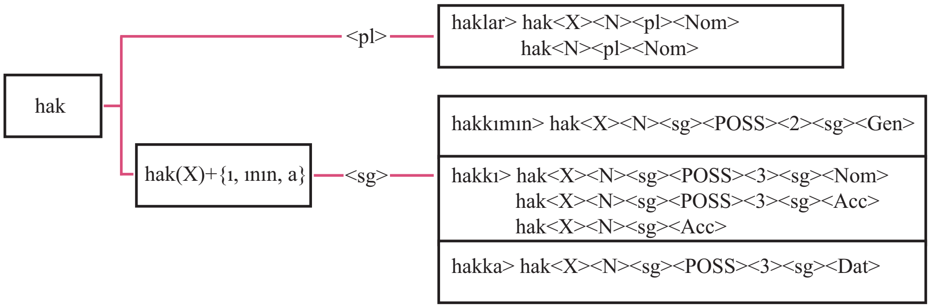
hak
<pl>
haklar> hak<X><N><pl><Nom>
hak<N><pl><Nom>
hak(X)+{ı, ının, a}
<sg>
hakkımın> hak<X><N><sg><POSS><2><sg><Gen>
hakkı> hak<X><N><sg><POSS><3><sg><Nom>
hak<X><N><sg><POSS><3><sg><Acc>
hak<X><N><sg><Acc>
hakka> hak<X><N><sg><POSS><3><sg><Dat>

Figure 5. Visualization of lexicon entry *hak* in singular and plural forms with its trigger function X.

4.2.7. Epenthesis

In some cases, a phoneme is added between two morphemes in order to make the pronunciation easier (epenthesis). This process is only partly predictable and is therefore marked in the lexicon with a trigger symbol [26].

If a noun ends with a vowel, for example *silgi* (‘eraser’), an /n/ is added before the genitive suffix -Yn. In other contexts, for nouns ending with a consonant, e.g., *kalem* (‘pencil’), a /y/ is added.

4.2.8. The aorist suffix

A further verbal suffix in Turkish is the aorist suffix -Yr. It behaves irregularly with respect to vowel harmony. Four cases can be distinguished. All polysyllabic verb stems and the following 12 monosyllabic verb stems fall into the first category: *al* (1take’), *ol* (1become’), *öl* (1die’), *gel* (1come’), *kal* (1remain’), *bul* (1find’), *var* (1have’), *ver* (1give’), *vur* (1beat’), *gör* (1see’), *san* (1suppose’), and *dur* (1stop’). This list of verbs was taken from the work of Öztaner [25]. These verbs follow the normal vowel harmony rules. The second category comprises all monosyllabic verb stems ending in a consonant, except those listed for the first category. If the vowel of the stem is a front vowel (see Table 8), the aorist suffix becomes -er; otherwise, it becomes -ar. The third category comprises verb stems that end with a vowel. In this case, the vowel of the aorist suffix is deleted. The final case concerns negated word forms. After the negation suffix -m‹K›, the aorist is always realized as /z/.

4.2.9. Passive

The Turkish passive suffix can realized in three ways depending on the context. If the stem ends with a vowel, it is realized as -n as in oku+n *okun* (‘to be read’), ara+n *aran* (‘to be searched/looked for’), or koru+n *korun* (‘to be protected’). If the stem ends with /l/, it is realized as -Yn, e.g., al+Yn *alın* (‘to be taken’), gül+Yn *gülün* (‘to be laughed’), or sil+Yn *silin* (‘to be wiped’), and in all other cases as -Yl. Some examples are yaz + Yl *yazıl* (‘to be written’), gör + Yl *görül* (‘to be seen’), and soy + Yl *soyul* (‘to be robbed’). We use the trigger ‹del› for vowel deletion for passive word forms.

5. Design of the gold standard

Morphological analyzers analyze morphemes. This means that only the grammatical categories are included, which are realized as morphemes. The gold standard of morpheme analyses includes the correct grammatical morpheme analyses, which were used as a reference in evaluation [27]. A gold standard (i.e. data that are manually annotated with the correct analyses) is an important part of any quantifiable evaluation. For our evaluation, we needed a list of word forms annotated with their correct morphological analyses. This raises the question of what the correct analysis of a complex word (homograph) such as *gözleme* (1. observation and 2. pancake) should be. We could analyze the whole word as a lexicalized form, or we could split it into smaller parts according to word formation rules [28]. TRMOR generates the following analyses for *gözleme*:

göz‹N›l‹K›m‹K›‹N›‹SUFF›‹sg›‹Nom›
gözle‹V›m‹K›‹V›‹SUFF›‹neg›‹imp›‹2›‹sg›
gözle‹V›m‹K›‹V›‹SUFF›‹neg›‹imp››2›‹sg›
gözle‹V›m‹‹N›‹SUFF›‹neg›‹POSS››3›‹sg›‹Dat›
gözle‹V›‹N›‹SUFF›‹POSS›‹3›‹sg›‹Dat›
gözle‹V›‹N›‹SUFF›‹sg›‹Dat›
gözlem‹N›‹sg›‹Dat›
gözlem‹N›‹sg›‹POSS›‹3›‹sg›‹Dat›
gözleme‹N›‹sg›‹Nom›

TRMOR provides analyses of varying granularity for the entry *gözleme* in four categories: the granularity of the analyses given in decreasing degrees, from the fine-grained analysis to the near representatives achieved. From *göz* ('eye'), a new semantically not very wrong word arises through the derivational suffix -lKmK. We decided to choose the right level of granularity if the analyses were close to the meaning. This means that if the word *hazır* ('ready'), for example, is analyzed in a fine-grained manner as *haz* ('enjoyment') + the transitive suffix -Yr, we have a wrong analysis. During this process, an interesting question arises of which granularity of the analysis should be obtained [20].

6. Evaluation

We evaluated TRMOR based on two large corpora and compared it to the TRmorph analyzer for Turkish in order to assess the quality and coverage of the system. During the development of TRMOR, we initially tested the rules with example words. Then a word list was extracted from an online newspaper[1] and nonwords and numbers were removed from the document. This evaluation raised the question of whether it is more important to increase coverage or to avoid overgeneration, since high coverage usually leads to more overgeneration.

Table 10 illustrates the frequency of a 1000-word Wikipedia list taken from [15] that is randomly extracted. The column "words in range" represents the number of words selected from the corresponding frequency range of the 1000-word Wikipedia list. For example, the first entry in Table 10 indicates that 159 words of the word list are selected from 16–20 of the frequency range.

Table 10. Distribution frequency on 1000 words of Wikipedia word list.

Frequency range	# words in range
16–20	159
21–65	476
66–100	103
101–25,801	262

6.1. Method

For the evaluation, a 2-million-word corpus with frequency information was used, which was extracted from the lexicon of Vikipedi (the Turkish version of Wikipedia), and a word list of the word types was extracted. Frequencies of 1 were deleted from the corpus and the evaluation was carried out afterwards. The gold standard on this body itself was produced semiautomatically. From 489,274 entries, which resulted after the deletion of frequency 1, 1000 entries determined at random from the 100th entries were selected. From these entries, some were the correct analyses, some were correct but did not produce all the correct analyses, some were wrong, and some had no analysis at all or could not be analyzed by TRMOR because the entry was not available in the TRMOR lexicon or it was a nonword, or else the word was wrong orthographically or grammatically in the Vikipedi corpus. All of these 1000 words were executed on the gold standard. In the absence of analyses, the possible analyses were completed and the wrong ones were corrected and completed.

[1] http://www.haber7.com/genel-saglik/haber/732846-cocuklari-istismardan-nasil-koruyabiliriz

6.2. Execution

During the execution of the gold standard, we handled not only the entries that were incorrectly written orthographically but also those grammatically. The evaluation was then executed on this file. In the second phase, unclear analyses were automatically deleted. All other analyses were then examined in detail. When processing 1000 word forms using TRMOR, 1343 analyses were obtained.

6.3. Result

Table 11 shows the calculation of precision and recall in order to evaluate all the expected analyses for each word form and thus each output comparing the output of the tool to one of four categories: true positive (candidate suitable and correct analysis), false positive (candidate suitable and wrong analysis or candidate not suitable and the tool does not provide any analysis), true negative (candidate not suitable and tool does not provide analysis), and false negative (candidate is suitable but analysis/analyses is/are not provided).

Table 11. Results of the evaluation on the Vikipedi test corpus.

	# of words	Precision %	Recall%	F-Measure %
Wikipedia	1.000	94.12	79.80	86.37

7. Comparison of two systems: TRMOR vs. TRmorph

We ran the TRMOR and TRmorph (2011 version) systems on the words extracted from Vikipedi. With the parsed words in Table 12, we gain all analyses given for an entry. This table shows the number of gold standard analyses generated from each system correctly (true positives), the number of incorrect analyses that each system generated (false positives), and the number of words that received no analysis (no results). For these experiments, a new list of 108 words was analyzed using both systems. Later, each analysis was manually classified as true or false. Due to the missing false negative numbers, we cannot calculate values of recall and F-measure. Therefore, an evaluation of TRmorph on the gold-standard data was not possible for the evaluation metrics because of differences in the analysis formats of TRMOR and TRmorph.

TRMOR gives 2.72 analyses approximately. However, TRmorph gives 4.53 analyses approximately per word with a minimum of one analysis using the Wikipedia word list. Since there is no negative category for morphological analysis, we only report the accuracy. The accuracy is 72% for TRMOR and 38% for TRmorph, as shown in Table 12. A quantitative comparison of coverage of the systems made for the analyzed and nonanalyzed entries respectively of TRMOR vs. TRmorph was determined.

Table 12. Comparison of TRMOR and TRmorph on word list.

Tools	Parsed words %	True positives %	False positives %	No result %
TRMOR	37	72	18	7
TRmorph	62	38	40	0.4

For these sets of words, the reason for failure was manually identified. Table 13 presents these words divided into two categories: analyzed and nonanalyzed. The nonanalyzed words are mostly names or technical terms. The same procedure was repeated for the correctly analyzed words, where no errors but only a few (erroneously) ambiguous analyses were obtained.

Table 13. Coverage of word list (total: 757).

	TRMOR	Trmorph	Both
Analyzed	1681	3041	487
Not analyzed	264	133	112

Overgenerating is a problem for morphological analyzers. For 108 entries, we get from TRMOR almost 10 times an entry, whereas this number in TRmorph is ca. 30 times. The derivation suffixes -mK and -lY are one reason for overgeneration. The correct verbal suffix in the word form is the obligative mood suffix -mKlY that can be followed by a personal morpheme or a copula morpheme.

8. Summary and future works

We have presented the TRMOR system, a finite-state morphological analyzer for Turkish that covers Turkish inflection, derivation, and some composition. The treatment of composition is interesting and difficult, since almost every declination of compound words differs from others. For the word *acemborusu* (the Turkish common name of *Bignonia radicans*), the system fails for the compound marker '*su*', except for the genitive marker for the nominative case, whereas for the word *ayçiçeği* ('sunflower'), 'i' is the compound marker. It seems that the compound markers occur differently according to the second noun, as in the first example for *boru* ('tube') and in the second example for *çiçek* ('flower'). Therefore, the implementation for all compound words will be handled in future works. The system analyzes the word forms, decomposes them into morphemes, and tests them for accuracy and correctness. Its accuracy and coverage were tested on real data (an online newspaper) and were compared to those of an existing analyzer. Besides having no cost, it is precise. However, the quantitative results of a system mean little if morphologically incorrect analyses are also accepted. TRMOR can be freely used and modified for various tasks including machine translation, named entity recognition, semantic analysis, or OCR error detection and it can serve as a basis for the creation of morphological word-form analyzers for related languages. TRMOR will be further expanded, improved, and applied to large corpora obtained from OCR data.

References

[1] Schmid H. A programming language for finite state transducers. In: FSMNL; Helsinki, Finland; 2005. pp. 308-309.

[2] Koehn P. Europarl: A parallel corpus for statistical machine translation. In: MT Summit; Edinburgh, UK; 2005. pp. 79-86.

[3] Eryiğit G, Oflazer K. Statistical dependency parsing of Turkish. In: 11th Conference of the European Chapter of the Association for Computational Linguistics; Trento, Italy; 2006. pp. 89-96.

[4] Göksel A, Kerslake C. Turkish: A Comprehensive Grammar. London, UK: Routledge, 2004.

[5] Can B. Unsupervised learning of allomorphs in Turkish. Turkish Journal of Electrical Engineering & Computer Sciences 2017; 25 (4): 3253-3260. doi: 10.3906/elk-1605-216

[6] Schmid H, Fitschen A, Heid U. SMOR: A German computational morphology covering derivation, composition and inflection. In: 4th International Conference on Language Resources and Evaluation; Lisbon, Portugal; 2004. pp. 1263-1266.

[7] Hankamer J. Turkish vowel epenthesis. In: Erguvanli-Taylan E, Rona B (editors). Puzzles of Languages: Essays in Honour of Karl Zimmer. Wiesbaden, Germany: Harrasowitz-Verlag, 2011, pp. 55-69.

[8] Antworth EL. PC-KIMMO: A two-level processor for morphological analysis. In: Summer Institute of Linguistics; Dallas, TX, USA; 1990.

[9] Beesley KR, Karttunen L. Finite-state morphology: Xerox tools and techniques. Stanford, CA, USA: CSLI, 2003.

[10] Hulden M. Foma: A finite-state compiler and library. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session; Athens, Greece; 2009. pp. 29-32.

[11] Lindén K, Silfverberg M, Pirinen T. HFST tools for morphology–an efficient open-source package for construction of morphological analyzers. In: International Workshop on Systems and Frameworks for Computational Morphology; Berlin, Germany; 2009. pp. 28-47.

[12] Oflazer K. Two-level description of Turkish morphology. Literary and Linguistic Computing 1994; 9 (2): 137-148.

[13] Hankamer J. Finite state morphology and left-to-right morphology. In: West Coast Conference on Formal Linguistics; Stanford, CA, USA; 1986.

[14] Karttunen L. Finite-state constraints. In: Goldsmith J (editor). The Last Phonological Rule 6. Chicago, IL, USA: University of Chicago Press, 1993, pp. 173-194.

[15] Çöltekin Ç. A freely available morphological analyzer for Turkish. In: 7th International Conference on Language Resources and Evaluation; Valetta, Malta; 2010. pp. 820-827.

[16] Springmann U, Schmid H, Najock D. LatMor: A Latin finite-state morphology encoding vowel quantity. Open Linguistics 2016; 2: 386–392.

[17] Karp D, Schabes Y, Zaidel M, Egedi D. A freely available wide coverage morphological analyzer for English. In: Proceedings of the 14th Conference on Computational Linguistics-Volume 3; Nantes, France; 1992. pp. 950-955.

[18] Cap F. Morphological processing of compounds for statistical machine translation. PhD, University of Stuttgart, Stuttgart, Germany, 2014.

[19] Karttunen L, Beesley KR. A short history of two-level morphology. In: ESSLLI-2001 Special Event Titled Twenty Years of Finite-State Morphology; Helsinki, Finland; 2001.

[20] Oflazer K, Inkelas S. A finite state pronunciation lexicon for Turkish. In: Proceedings of the EACL Workshop on Finite State Methods in NLP; 2003. pp. 900-918.

[21] Solak A, Oflazer K. Parsing agglutinative word structures and its application to spelling checking for Turkish. In: 1992 Proceedings of the 14th Conference on Computational Linguistics-Volume 1; Nantes, France; 1992. pp. 39-45

[22] Kaplan RM, Kay M. Regular models of phonological rule systems. Computational Linguistics 1994, 20: 331-378.

[23] Fitschen A. Ein computerlinguistisches Lexikon als komplexes System. PhD, University of Stuttgart, Stuttgart, Germany, 2004 (in German).

[24] Van der Hulst H, Van De Weijer J. Topics in Turkish phonology. In: Boeschoten H, Verhoeven L (editors). Turkish Linguistics Today. Leiden, the Netherlands: E.J. Brill, 1991, pp. 11-59.

[25] Öztaner SM. A word grammar of Turkish with morphophonemic rules. MSc, Middle East Technical University, Ankara, Turkey, 1996.

[26] Pado S. Computerlinguistik und Sprachtechnologie: Eine Einführung. 3rd ed. Berlin, Germany: Springer-Verlag, 2009 (in German).

[27] Kurimo M, Creutz M, Varjokallio M. Unsupervised morpheme analysis evaluation by a comparison to a linguistic gold standard. In: CLEF; Budapest, Hungary; 2007.

[28] Faaß G, Heid U, Schmid H. Design and application of a gold standard for morphological analysis: SMOR as an example of morphological evaluation. In: 7th International Conference on Language Resources and Evaluation; Valetta, Malta; 2010. pp. 803-810.