Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3984–3990
Marseille, 11–16 May 2020

Visual Modeling of Turkish Morphology

Berke Özenç, Ercan Solak
Department of Computer Engineering
Işık University, İstanbul, Turkey
{berke.ozenc,ercan.solak}@isikun.edu.tr

Abstract

In this paper, we describe the steps in a visual modeling of Turkish morphology using diagramming tools. We aimed to make modeling easier and more maintainable while automating much of the code generation. We released the resulting analyzer, MorTur, and the diagram conversion tool, DiaMor as free, open-source utilities. MorTur analyzer is also publicly available on its web page as a web service. MorTur and DiaMor are part of our ongoing efforts in building a set of natural language processing tools for Turkic languages under a consistent framework.

Keywords: Turkish, morphology, morphological analyzer

1. Introduction

Morphological analysis is among the first steps in the natural processing pipeline of morphologically rich languages. Analysis is often preceded by the relatively simpler steps of sentence segmentation and tokenization. Morphological analysis of a token usually yields more than one analysis, in which case a disambiguation step is performed before any further processing. Disambiguation is closely related to syntactic parsing as the distinct analyses yield the first level nodes in distinct parses for a given sentence.

The accuracy of an analyzer depends on both the extent of its base lexicon and the fidelity of its morphological/phonological model of the target language. Constructing a base lexicon and a model requires linguistic expertise in the target language and is a labor intensive tasks. In this paper, we demonstrate how the modeling stage can be made easier using visual tools and automated code generation.

We used the diagramming tool draw.io (draw.io, 2019) to completely model the morphotactics of Turkish in a diagram and we automatically converted the diagram into a finite-state transducer (FST) model using DiaMor (Özenç and Solak, 2019c). We provide in the following sections, the details of the model, its linguistic base and its tag set. The resulting analyzer, MorTur, is freely available together with its source.

Currently, there are three publicly accessible morphological analyzers for Turkish, (Akın and Akın, 2018), (Şahin et al., 2013), (Çöltekin, 2014). Although the first computational description of Turkish morphology was given in (Oflazer, 1994), Oflazer's `xfst` based implementation is not publicly accessible as of December 2019.

ITU analyzer described in (Şahin et al., 2013) is based on Oflazer's description in (Oflazer, 1994) and closely follows its conventions and tags. The analyzer's source code is not publicly available. Still, it can be used through its web page for small amounts of text. It is also possible to use the analyzer as a web service upon permission from its creators.

The Zemberek analyzer is an open source java tool, (Akın and Akın, 2018). The implementation is done on a custom implementation of a finite transducer engine. The analyzer can be embedded as part of a larger java codebase.

TrMorph analyzer (Çöltekin, 2014) is implemented on `foma` (Hulden, 2009) and its source code is available on `github`. TrMorph can be used on its web page for single token queries.

All three analyzers, ITU, Zemberek and TrMorph, may yield multiple analyses for a given surface form. However, the number of analyses they return differs. For example the word *dolardır* (it is dollar) yields 4, 5 and 17 analyses for the three analyzers, ITU, Zemberek and TrMorph, respectively.

Our motivation for creating a new analyzer is two-fold. First, we wanted to build a free and open-source analyzer that has a linguistically consistent coverage. The differences in the number of analyses in existing analyzers indicate such a need. We explain this point further in Section 4.. Second, we wanted to build a visual model that can be easily modified and ported to build analyzers for other major Turkic languages. To this date, apart from Turkish, we have finished implementations for MorAz, an analyzer for Azerbaijani, the language that is spoken in Azerbaijan and Iran (Özenç et al., 2018), (Özenç et al., 2017). In the near future, we are planning to implement analyzers for Kazakh, Uzbek, Kyrgyz and Turkmen. Currently, there are few analyzers for these languages with varying degrees of coverage (Kessikbayeva and Cicekli, 2016), (Matlatipov and Vetulani, 2009), (Washington et al., 2012),(Tantug et al., 2006).

The organization of the paper is as follows. In Section 2. we briefly describe salient properties of Turkish morphology and phonology. In Section 3., we describe the tag set and the output format of MorTur. In Section 4., we lay down the linguistic foundations underlying our approach. In Section 5., we provide the details on the visual modeling and implementation. Section 7. reports some statistics comparing MorTur to other available analyzers.

2. Turkish

Turkish is an agglutinative, right-headed language with a relatively free word order. The choice of a particular word order is largely governed by the pragmatics of the discourse. Alongside the word order, prosody markers such as stress, pause and intonation indicate the pragmatic aspects such as focusing and backgrounding, (Erguvanli and Taylan, 1984).

Modern Turkish lexicon includes historical borrowings from French, Arabic and Persian. These borrowed words have often undergone changes that project them onto the sound system of Turkish phonology. Still, some borrowed roots in the lexicon carry their own phonological constraints which need to be taken into account in implementing the phonology module of a morphological analyzer. This issue is detailed further in Section 6..

Turkish morphology has a complex yet mostly regular morphotactics with a fixed order of morphemes with a notable exception concerning the interaction of the 3rd Person Plural morpheme -lAr with its surrounding context (Özenç and Solak, 2019b).

Turkish has a rich derivational morphology with varying degrees of productivity. Often, the semantics of a derivation can be compositionally inferred as in the cases of Verbal participles and Adverbial derivations. On the other hand, for Nominal derivations such as -CI, although the morpheme is productive, its precise semantics cannot be predicted by the composition alone.

Syntactic relations in a sentence in Turkish are specified by both the word order and the case marking. Adverbial adjuncts can be shuffled (up to prosodic emphasis) among themselves without a change in the underlying semantics. Similarly, Adjectives qualifying the same noun phrase can be scrambled.

The phonology of Turkish has back/front and rounded/unrounded vowel harmony, devoicing at the end of morphemes and epenthesis at the start of morphemes. It also has syncope in the last vowels of few roots, mostly borrowed from Arabic. Devoicing at the end of the roots has some irregularities that need to be marked in the lexicon.

3. Analysis Format

The analysis format in MorTur is basically a root morpheme followed by a sequence of abstract morphemes and tags. An example analysis with its corresponding surface form is given in (1).

(1) gel<VS><Pol:Neg><Tns:Past><Prsn:1s>
gel-me-di-m
I did not come

Category tags like <VS> specify the morphological category of the whole stem that is to its left. Category tags do not have a corresponding surface form. Note that our category tags are different than POS tags. We explain our rationale for using such category tags in Section 4..

Tags like <AgtA> (aorist agent) represent derivational abstract morphemes. They are always followed by category tags denoting the category of the derived stem as in (2).

(2) gel<VS><Pol:Neg><AgtA><NOM>
gel-me-yen
the one who does not come

Key-value tags like <Prsn:1s> are used for inflectional abstract morphemes. The key denotes the function of the morpheme. For example, in <Prsn:1s>, the mnemonic key `Prsn` indicates that the morpheme function is to denote Person in verb inflection. The full list of abstract morphemes and category tags together with their corresponding Leipzig tags is given in (Özenç and Solak, 2018). A screenshot of the web analyzer interface is given in Figure 1.

4. Linguistic Foundations

MorTur differs from other analyzers for Turkish in a few fundamental aspects. In this section we briefly discuss our rationale for these differences.

4.1. Lexicon

The choice of root lexicon has a major effect on the number of distinct analyses a word form yields, (Ehsani et al., 2018). At one extreme, all the derived forms can be viewed as part of the lexicon and can be included in a dictionary. This extreme choice explodes the number of redundant analyses to the extent that all the derivations are included together with the root morphemes.

The other extreme is to start with a minimal lexicon that includes only the word forms that do not have any suffixes. Such an approach treats all derivations as productive and over-generates word forms.

In our approach in designing MorTur, we chose a middle ground between these two extremes, at the expense of some manual data preparation. We started with the list of single token lemmas in the official Contemporary Dictionary of Turkish (CDT), (TDK, 2019). CDT, by its construction, leans towards the first extreme and includes many productive derivations. We manually pruned the list with the help of a tool we constructed for this purpose. For all the derived lemmas in the dictionary, we checked whether the semantics of the derived lemma can compositionally be derived from the semantics of its shorter stem (which is already included in the dictionary) and the semantics of the derivational morphemes generating the derived lemma. For example, the lemma *kırmızı-laş-mak* (to become red) is a lemma in CDT. However, its semantics can be compositionally derived from the semantics of the already listed lemma *kırmızı* (red) and those of the derivational morphemes *-lAş* (become, transform into) and *-mAk* (infinitive suffix). Therefore, we prune the lemma *kırmızılaşmak* out of the initial lexicon.

After the manual elimination, the root lexicon has 2957 verb roots, 28246 nominal roots and 1391 adverbial roots. Thus, the root lexicon of MorTur is minimal in the sense that analyses do not contain roots and their productive derivations at the same time.

4.2. Morphological Categories

In Turkish, as in other languages, the POS tag of a word form is conditioned by its surrounding context. On the other hand, in Turkish, the morphological behavior of a stem is conditioned by its sequence of morphemes and does not depend on the context. For example, *yavaş* (slow) behaves like a Noun morphologically and can therefore be suffixed like a Noun as in (3), where we omitted our category tags and the zero morphemes for the sake of brevity.

Figure 1: The screenshot of MorTur web interface.

(3) yavaş-ı vur-du-m
slow<Case:Acc> hit<Tns:Past><Prsn:1s>
I hit the slow one

At the same time, the root *yavaş* can function as an Adjective as in *yavaş trafik* (slow traffic) or an Adverb as in *yavaş git* (go slowly).
In MorTur, we distinguish between the syntactic POS and the morphological category. Thus, everything that behaves like a Noun morphologically is categorized as <NOM> (Nominal). This approach reduces the number of analyses, separates the semantics from morphology and also alleviates the need for zero morphemes. With this distinction, the POS decision is deferred to the syntactic parsing stage where the analyses of the surrounding word forms are processed together.

4.3. New Morphemes

In other analyzers for Turkish, as well as in traditional approaches to Turkish morphology, the participle morphemes are treated as a class of their own, mainly due to their prominent use in relative clause constructions (Kornfilt, 1997). In MorTur, we treat participles as derivational morphemes that derive Nominals (NOM) out of verb stems (VS). In this view, participles are no different in their behavior than other derivational morphemes. The three most common participles in Turkish are -(y)An, -DIk and -(y)AcAk. In MorTur, we represent them as derivational abstract morphemes as <AgtA> (the agent who performs the action in Aorist tense), <InfP> (the infinitive semantics of the action in past tense) and <InfF> (the infinitive semantics of the action in future tense), respectively. They are followed by the <NOM> category tag in the analyses. For a detailed discussion on the treatment of participles as derivational morphemes and its relation to the syntax of relative clauses, see (Solak, 2019).

Another new morpheme we introduce in MorTur is the Noun Compounding morpheme <Nc> which has the same surface form -(s)I(n) as that of the third person possessive morpheme <Poss:3s>, (Göksel and Kerslake, 2004). Noun compounding morpheme has also been compared to Persian ezafe, (Kharytonava, 2009). The noun compounding morpheme is one of the most common suffixes in Turkish, appearing in indefinite noun compounds like *otobüs bilet-i* (bus ticket) and *ev kapı-sı* (house door). Because of their identical surface forms, this morpheme is analyzed as 3rd Person Possessive suffix in other three analyzers. The distinction is important as a feature in syntactic parsing down the processing pipeline.

5. Implementation

We implemented the two-level morphology in MorTur on top of `HFST`, (Lindén et al., 2011; Beesley and Karttunen, 2003). The web service is provided with wrappers written in `python`. The source code is available on `github` and the detailed documentation is given on MorTur web pages, (Özenç and Solak, 2018). The web page also contains an interactive interface where it is possible to analyze multiple tokens in a single query.

The first level implements the morphotactics and the second level implements the phonology. Some phonological rules are embedded in the first level by way of choosing appropriate archmorphemes. For example, for some state transitions in the first level the epenthesis is dropped.

The input to the first level is a root morpheme followed by a sequence of abstract morphemes and category tags. Its output is a sequence of archmorphemes that are written in the alphabet of archiphonemes and optional epentheses. The first level output is transformed in the second level by a series of rewrite rules, each of which represents a particular process of phonology such as devoicing and vowel harmony. In (4), the first level input and the outputs of the first and the second levels are given for *geleceksin* (you will come).

(4) gel<VS><Tns:Fut><Prs:2s>
gel-(y)AcAk-sIn
gel-ecek-sin

The underlying FST can be run in either generation or analysis mode. In MorTur web interface, the FST is run in the analysis mode and the web application runs a small wrapper script in `python` to format the output and clean special marks in the lexicon.

The morphotactics of MorTur has 109 states. The full state diagram is given in the supplemental materials. Here, we illustrate in Figure 2 part of the Voice feature of the verb inflection paradigm.

In Figure 2, VC represents the state where the verb stems land after a series of inflections and derivations. Actually, VC represents a collections of states, each of which corresponds to phonological class of verb stems for an immediate Passive and Causative suffix. APP state represents the exit path from the Voice paradigm which is followed by the part of the FST where the morphotactics of Ability, Polarity and Probability (APP) are implemented. The 5 voices of the verb paradigm are Active, Passive, Reflexive, Reciprocal and Causative, the last of which might occur multiple times.

In our implementation we opted not to use flag diacritics although HFST implements them. The flag diacritics enable a concise expression of conditional suffixation. However it somewhat obscures the boundary between the morphology and semantics, which we tried to keep separate in MorTur. In any case, anything that is expressed with flag diacritics can also be expressed without them albeit at the cost of using more states. In our development, we find that the representation without the flag diacritics is more transparent for further modification of the morphotactics.

In order to simplify the modeling of morphotactics, we used the diagramming software `draw.io` to draw the full morphotactics and used DiaMor (Özenç and Solak, 2019c), (Özenç and Solak, 2019a) to automatically generate the FST code and compile the final analyzer. The complete diagram of the whole morphotactics that is input to DiaMor is provided in the supplementary materials.

6. Phonology

In Turkish, the vowel harmony in suffixes is parameterized by the backness and backness-roundedness. The choice of "a" or "e" is done by the backness of the preceding vowel and the choice of "ı", "i","u" or "ü" is done by the backness-roundedness combination of the preceding vowel. There are some exceptions like "-(I)mtrak" some of whose vowels do not undergo vowel harmony. For those that follow the vowel harmony, we use the usual archphonemes "A" and "I".

The stem final consonant "t" is voiced to "d" when followed by a vowel. However, this behavior is suppressed in many roots that are usually borrowed from Arabic. For example, in "devlet-i" (his/her state) voicing is suppressed but in "*kanat-ı", "kanad-ı", it is not suppressed. Since there is no obvious phonological rule that will predict the suppression, we manually marked those roots with special symbols in the lexicon. We construct the phonological re-write rules to recognize those symbols and suppress the voicing in the implementation.

Similarly, there are some phonological processes whose application cannot be predicted by their context and we marked those in the lexicon and implemented their corresponding re-write exceptions. Among the marked phenomena are the syncope of the penultimate vowel in some roots when followed by a vowel (e.g. boyun, boyn-um), the duplication of the final consonant in some Arabic-origin roots when followed by a vowel, (e.g. hak, hakk-ım), semi-fronted pronunciation of some back vowels in some roots, (e.g. alkol, alkol-ü).

Another common phonological parameterization occurs in the choice of the particular Causative (-DIr, -(I)t, -(I)r or -(A)r) or Passive (-(I)n or -(I)l) morpheme following a verb root. We marked this variation in the lexicon as a set of distinct initial states corresponding to each combination of paramterization. Figure 3, shows two such initial states and their immediately following states in Voice inflection. For example, the verb root "bul" (find) starts from the DIr_In_CAT state in Figure 3 and follows with "bul-un" or "bul-dur" for Passive and Causative, respectively.

Considering the possibility of 4 Causative morphemes and 2 Passive morphemes, we would expect to have 8 distinct initial states for verb stems. However, there a few stems that lack Causative or Passive forms or both. For example, the verb root "ağrı-" (ache) lacks the Passive form but has a Causative form "ağrı-t" (cause to ache). Hence, the number of possible initial Voice states becomes 15. Actually, of 2957 verb roots, 12 lack the Causative form, 4 lack the Passive and 7 lack both forms. These lead to 13 initial verb stem states instead of theoretical 15, due to two missing phonetic combinations.

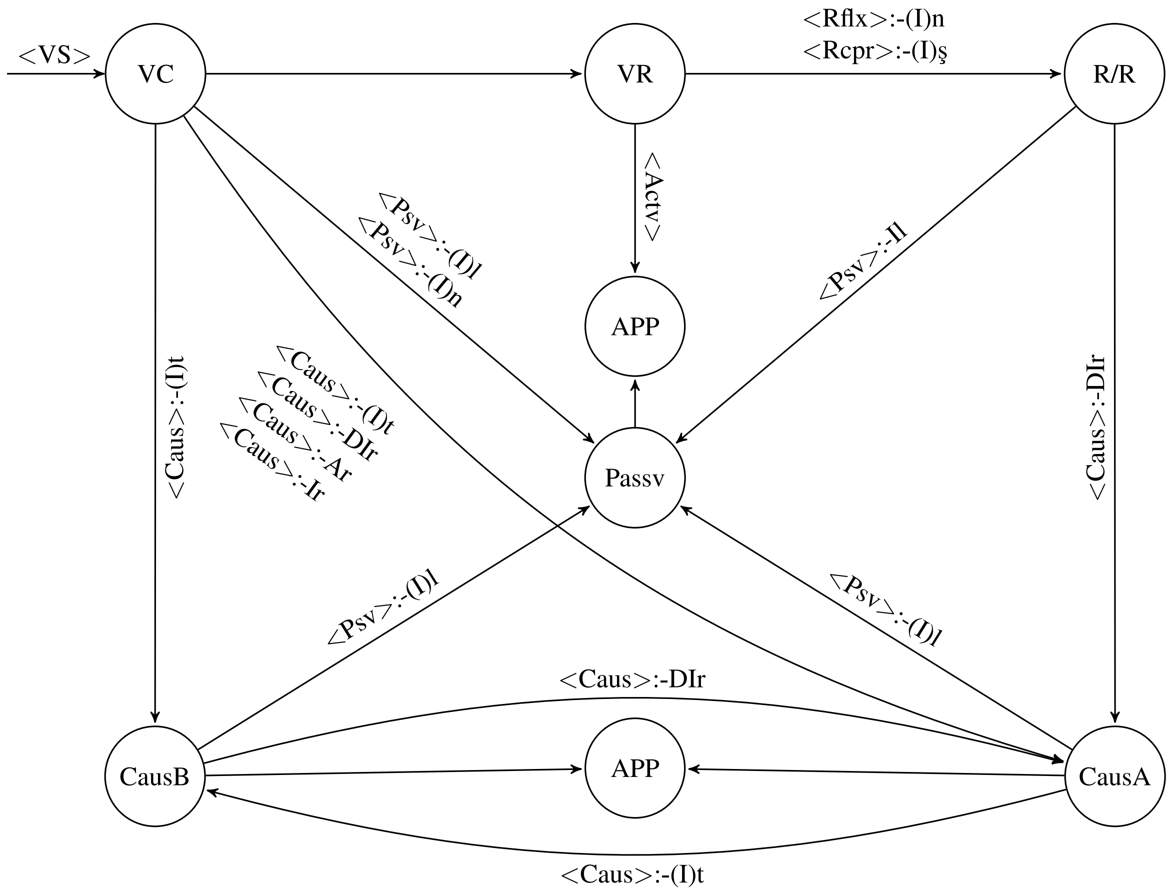

Figure 2: Voice part of the verb inflection in MorTur.

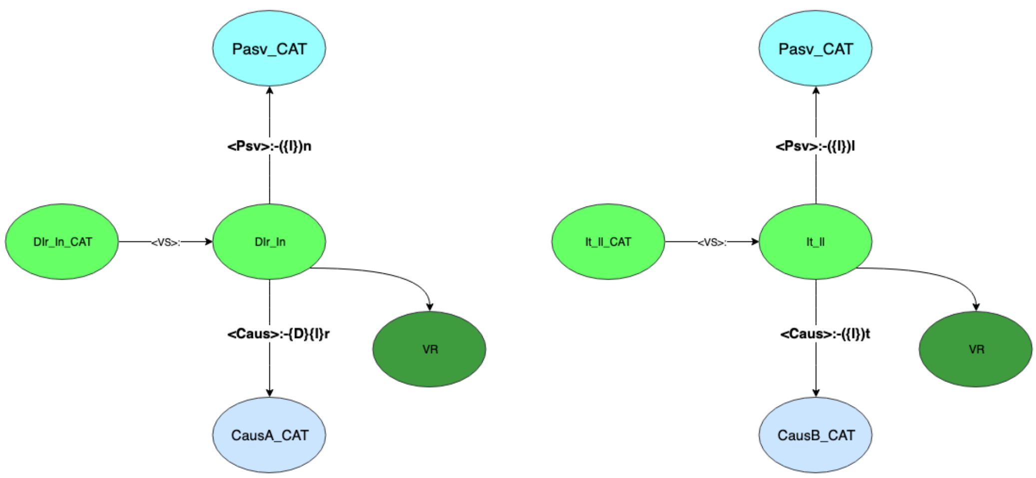

Figure 3: The partial diagram of the Voice in verb inflection.

7. Comparison

In this section we give the results of the comparisons between MorTur and the other publicly accessible analyzers as well as between MorTur and human annotators.
We picked 10K random word forms from the Milliyet corpus for comparing MorTur with other analyzers. We excluded the punctuation marks and proper names.
Table 1 summarizes the comparative statistics for the number of word forms not analyzed and the average number of analyses calculated over the word forms that are analyzed.

In Table 1, the percentage of word forms without an analysis is close to the ones obtained from the other analyzers. Out of the 67 word forms not analyzed by MorTur, 21 are different inflected forms of the question particle -mI. In MorTur, we treat the Question particle as part of the word it is suffixed to since it changes for the vowel harmony of the preceding stem and it is inserted before the Tense or Person suffixes in some verb inflection sub-paradigms, (Özenç and Solak, 2019b). Yet, the standard Turkish orthography requires a space between the stem and the Question particle. In MorTur, the input word forms are assumed to have been tokenized while keeping the inflected Question particle and its preceding stem as a single token. In the test corpus, -mI particle and its inflected forms like -mIydI are given as distinct tokens which are analyzed as such by the three analyzers. Discounting the missing 21 analyses for those, our miss rate would be 46 out of 10K, which is on par with the best coverage provided by TrMorph. There are also some misspelled word forms in the test corpus but they are equally missed by all the analyzers so this does not have any effect on the comparison.

For human annotator comparison, we randomly picked 100 words from the previous list of 10K words and had them annotated by three native speakers who have not worked in the development of MorTur. In order to simplify the annotation task for the native speakers who are not familiar with MorTur's tag set and the annotation conventions, we asked them to mark the morpheme boundaries for each analysis with a free text explanation on the semantics. We then manually converted these into the formal analyses in MorTur's format for a proper comparison. For each word form, we took the union of analyses found by human annotators.

MorTur yields 258 analyses in total for the whole list of 100 words, while the human annotators jointly found 147 analyses. All of these 147 analyses are among the ones found by MorTur. Thus, the recall rate is 100%.

The calculation of precision is a bit tricky. Human annotators tend to focus more on the most common analyses and miss the less obvious ones, although they realize what they miss once it is pointed out. For example, when asked to analyze the single token "bile", they identify the Conjunction or Adverb analyses but miss the verbal analyses given in (5).

(5) bil<VS><Pol:Pos><Tns:Opt><Prsn:3s>
let him/her know!
bile<VS><Pol:Pos><Tns:Imp><Prsn:2s>
sharpen!

Such correct analyses found by MorTur but missed by annotators thus lowers the precision score. Therefore, calculating precision scores of morphological analyzers against human annotators is not a very meaningful measure of the analyzers' performances.

8. Conclusion

In this paper, we gave the details of MorTur, a new open source morphological analyzer for Turkish and our visual modeling of its morphotactics using DiaMor. We built MorTur on a manually marked minimal root lexicon and a concise set of morphology tags. We generated the FST description for morphotactics through an automated parse of the diagram XML. Any further modification to the diagram is reflected in the FST which helps the maintenance of the analyzer through different versions. Furthermore, parts the visual diagram can be reused with some minor modifications in modeling other Turkic languages.

In MorTur, we focused on the morphological behavior of stems rather than their POS categories. We therefore combined the POS categories of Adjective and Noun under the morphological category Nominal. This approach eliminates the use of null derivational morpheme between Adjectives and Nouns. Thus, we defer the POS decision down to the parsing pipeline.

MorTur is part of our ongoing efforts in creating a set of computational morphology and syntax tools for the most popular Turkic languages. Up to now, we have finished implementing major parts of Azerbaijani. Both analyzers as well as the DiaMor tool are freely available on their respective web pages.

9. Bibliographical References

Akın, A. A. and Akın, M. D. (2018). Zemberek. `https://github.com/ahmetaa/zemberek-nlp`. [Online; accessed 10-April-2019].

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications,, Stanford, CA.

Çöltekin, c. (2014). A set of open source tools for Turkish natural language processing. In Nicoletta Calzolari, et al., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1079–1086. European Language Resources Association (ELRA).

draw.io. (2019). draw.io. `https://draw.io`. [Online; accessed 2-December-2019].

Ehsani, R., Solak, E., and Yildiz, O. T. (2018). Constructing a wordnet for Turkish using manual and automatic annotation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 17(3):24:1–24:15, April.

Erguvanli, E. and Taylan, E. (1984). *The Function of Word Order in Turkish Grammar*. Number 106-107. c.ler in The Function of Word Order in Turkish Grammar. University of California Press.

Göksel, A. and Kerslake, C. (2004). *Turkish: A Comprehensive Grammar*. Routledge, London.

Hulden, M. (2009). Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32. Association for Computational Linguistics.

Kessikbayeva, G. and Cicekli, I. (2016). A Rule Based Morphological Analyzer and a Morphological Disambiguator for Kazakh Language. *Linguistics and Literature Studies*, 4(1):96–104.

Kharytonava, O. (2009). Mots composées en Turc dans l'approche de la morphologie distribuée. In *Proceedings of the 2009 annual conference of the Canadian Linguistic Association*.

Kornfilt, J. (1997). On the syntax and morphology of relative clauses in Turkish. In *Dilbilim Araştırmaları*, pages 24–51.

Lindén, K., Axelson, E., Hardwick, S., Silfverberg, M., and Pirinen, T. (2011). HFST–framework for compiling and applying morphologies. In *Proceedings of Second International Workshop on Systems and Frameworks for Computational Morphology*, pages 67–85.

Matlatipov, G. and Vetulani, Z., (2009). *Representation of Uzbek Morphology in Prolog*, pages 83–110. Springer Berlin Heidelberg, Berlin, Heidelberg.

	MorTur	ITU	Zemberek	TrMorph
Not analyzed %	0.007	0.007	0.003	0.005
Average analysis count	2.62	2.18	2.49	3.28

Table 1: Comparative statistics for MorTur, ITU, Zemberek and TrMorph.

Oflazer, K. (1994). Two-level description of Turkish morphology. *Literary and Linguistic Computing,*, 9(2):137–148.

Özenç, B. and Solak, E. (2018). MorTur. `http://ddil.isikun.edu.tr/mortur/`. [Online; accessed 10-April-2019].

Özenç, B. and Solak, E. (2019a). Diamor. `https://github.com/berkeozenc/Diamor`. [Online; accessed 2-December-2019].

Özenç, B. and Solak, E. (2019b). A systematic analysis into the morphotactics of Person, Question and multiple Tenses in Turkish. In *Proceedings of the VII International Conference on Computer Processing of Turkic Languages*, Simferopol, Crimea, Russia, October.

Özenç, B. and Solak, E. (2019c). Visual modelling of morphology. In *Proceedings of the VII International Conference on Computer Processing of Turkic Languages*, Simferopol, Crimea, Russia, October.

Özenç, B., Ehsani, R., and Solak, E. (2017). MorAz. `http://ddil.isikun.edu.tr/moraz/`. [Online; accessed 10-April-2019].

Özenç, B., Ehsani, R., and Solak, E. (2018). Moraz: an open-source morphological analyzer for Azerbaijani Turkish. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–29, Brussels, Belgium, November. Association for Computational Linguistics.

Şahin, M., Sulubacak, U., and Eryiğit, G. (2013). Redefinition of Turkish morphology using flag diacritics. In *Proceedings of The Tenth Symposium on Natural Language Processing (SNLP-2013)*, Phuket, Thailand, October.

Solak, E. (2019). On relative clause in Turkish. In *Proceedings of the VII International Conference on Computer Processing of Turkic Languages*, Simferopol, Crimea, Russia, October.

Tantug, A. C., Adali, E., and Oflazer, K. (2006). Computer Analysis of the Turkmen Language Morphology. *FinTAL*, 4139:186–193.

TDK. (2019). Güncel Türkçe Sözlük. `https://sozluk.gov.tr/`. [Online; accessed 2-December-2019].

Washington, J., Ipasov, M., and Tyers, F. (2012). A finite-state morphological transducer for Kyrgyz. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 934–940, Istanbul, Turkey, May. European Language Resources Association (ELRA).