Turkish NLP web services in the WebLicht environment

Çağrı Çöltekin
University of Tübingen
Wilhelmstr. 19, 72074 Tübingen
ccoltekin@sfs.uni-tuebingen.de

Abstract

This document introduces a number of Turkish natural language processing tools that are being integrated into the CLARIN infrastructure. We describe the tools and resources used in this effort, present their evaluation results, and discuss particular challenges met during this effort both due to some properties of the language and the available resources.

1 Introduction

Turkish is a language spoken mainly in Turkey by about 70 million people. It is also one of the major immigrant languages in Europe, e.g., by over 2 million speakers in Germany. Turkish is typologically different from the languages that are well-represented by a large number of tools and resources in the CLARIN infrastructure, and it poses some unusual challenges for the theory and the practice of linguistic research. Hence, besides their practical value because of the large number of speakers, the natural language processing services for Turkish may also be of interest for theoretical (linguistic) research. The services introduced in this document are useful for linguistics research, and potentially for other areas of humanities research.

The main difficulties or differences in computational processing of Turkish are related to its morphological complexity. Morphological complexity, in case of Turkish, does not only mean that the words in the language have a rich set of morphological features. Some linguistic functions that are typically realized in syntax in other languages, e.g., subordinate clause constructions, are realized in morphology in Turkish. As a result, the computational linguistic methods and tools that assume whole words as the units in syntax have a number of difficulties processing Turkish. Example (1) demonstrates one of these problems.

(1) *Sorun* *tarafların* *konuşmamasıydı*
Problem side-PL-GEN talk-NEG-INF-POSS3P-PAST-PERS3P

'The problem was (the fact that) the parties did not talk.'

Besides the diverse list of morphological features assigned to the last word in (1) that would not fit into a single morphological paradigm, it is clear from the English translation that there are two different predicates in this example ('be' and 'talk'), both having subjects of their own. However, the Turkish sentence does not have two separate words for each predicate. Both predicates are confined into a single word, *konuşmamasıydı*. Furthermore, the negation clearly belongs to the verb *konuş-* 'talk', not to the copula, and the past tense marker belongs to the copula ('was'). As a result, the proper analysis of such sentences requires syntactic relationships between parts of the words, and hence, presenting challenges to typical computational linguistic methods which assume the word is the minimal syntactic unit. In the remainder of this document we describe the way we fit a set of Turkish NLP tools to allow morphological analysis/disambiguation and dependency parsing within the WebLicht environment.

2 The Turkish NLP tools in the WebLicht environment

WebLicht (E. Hinrichs et al. 2010; M. Hinrichs et al. 2010) is a natural language processing environment that enables researchers to use NLP web services offered by a large number of institutions. WebLicht

allows chaining these services in custom ways to obtain the desired linguistic annotations, and visualize the results through a user-friendly web-based interface. A number of different NLP tasks, e.g., tokenization, POS tagging, dependency or constituency analysis, for a number of languages (including German, English, Dutch, French, Italian) are readily available in the WebLicht environment. WebLicht enables researchers without substantial computational experience to make use of these automated tools. WebLicht is developed within the CLARIN project, and it is fully integrated to the rest of the CLARIN infrastructure. In this section, we describe the new Turkish NLP web services that are being integrated to the WebLicht environment. Some of the tools described here are based on existing tools and resources, and some of them are developed from scratch or improved substantially during the process of integrating them to WebLicht. Although similar efforts exist (most notably Eryiğit 2014), our approach differs in the choice of tools in the pipeline, and integration into the WebLicht provides an easy-to-use and familiar system for the users of the CLARIN infrastructure.

2.1 Sentence and word tokenization

Sentence and word tokenization is typically the first task in an NLP pipeline. Since Turkish is written with a Latin-based alphabet, this task is similar to tokenization of most European languages. For both tokenization tasks, we modify existing tokenization services based on Apache OpenNLP, and add statistical models for Turkish. The sentence splitter model is trained using 1 million sentences from the Turkish news section of the Leipzig corpora collection (Quasthoff et al. 2014). The F_1-score of the resulting sentence splitter on the same corpus is $95.8\,\%$ (average of 10-fold-cross validation, sd=$0.000\,5$). A qualitative analysis of the results indicates that a sizable part of the mismatch between the model's output and the gold standard is not due to errors made by the model, but errors in the original automatic sentence tokenization. The F-score goes up to $98.7\,\%$ if the model is trained on full Leipzig corpus and tested on about five thousand sentences from the METU-Sabancı treebank (Say et al. 2002).

As noted in Section 1, words are not the tokens that are used for some of the NLP tasks, especially for parsing. The tokens that are input to the parsing can only be determined after the morphological analysis. However, we still need a first tokenization pass for other NLP tasks, including morphological analysis. We train the OpenNLP tokenizer on the METU-Sabancı treebank, which results in a model with $0.44\,\%$ error rate (average of 10-fold cross validation).

2.2 Morphological analysis and disambiguation

The morphologically complex nature of the language puts morphological analysis on a central place in Turkish NLP. For morphological analysis, we use the open-source finite-state morphological analyzer TRmorph (Çöltekin 2010; Çöltekin 2014). Besides the morphological analyzer, TRmorph distribution contains a guesser which is useful if the root of the word is unknown. The output of the morphological analyzer for the verb in Example 1 is given in (2).

(2) `konuş⟨V⟩⟨neg⟩⟨vn:inf⟩⟨N⟩⟨p3s⟩⟨0⟩⟨V⟩⟨cpl:past⟩⟨3s⟩`

It is important to note, for the purposes of this work, that the analysis contains multiple part of speech tags. That is, the root *konuş* 'talk' is a verb which is inflected for negation, then the verb becomes a noun (nominal) with the meaning 'the state of not talking', and it again becomes a verb (a nominal predicate). This process is rather different than usual morphological derivation. The crucial difference is that each step in this derivation may participate in syntactic relations outside the word (see in Section 2.3 for an example). The conventional solution for analyzing such words in Turkish NLP involves assuming subword syntactic units called *inflectional groups* (IG). An IG contains a root or a derivational morpheme and a set of inflectional features, or inflections.[1] Each IG may potentially participate in syntactic relations with other IGs outside the word. As well as determining the inflections, or morphological features, of each IG, identifying these IGs is also part of the morphological analysis. Hence, morphological analysis also functions as a tokenization step for the syntactic analysis. For use in WebLicht, we have implemented

[1] This definition is slightly different than earlier use in the literature (e.g., Hakkani-Tür et al. 2002), where derivational morphemes were considered as part of the 'inflectional features'.

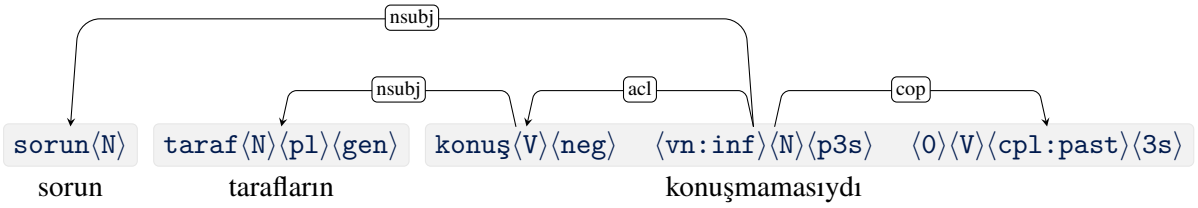

Figure 1: A dependency analysis of Example 1. The dependency labels (roughly) follow Universal Dependencies (http://universaldependencies.github.io/docs/). Dependency annotation produced by our parsing service currently follows the METU-Sabancı treebank format which is slightly less intelligible. The morphological tags/labels are from TRmorph.

a simple finite-state decoder that uses the finite-state transducer produced by TRmorph description. For each word, the transducer output is converted to a set of IGs with a POS tag, and a set of morphological features relevant for that POS tag.

Although finite-state tools are efficient at producing analyses strings like in (2), the analyses are often ambiguous. For example, the first word *sorun* in (1), among a few others, can also be analyzed as sor⟨V⟩⟨imp⟩⟨2p⟩ '(please) ask!' or soru⟨N⟩⟨p2s⟩ 'your question'. Disambiguation of these analyses is important, and it has attracted considerable interest in Turkish NLP literature (Hakkani-Tür et al. 2002; Yüret and Türe 2006; Sak et al. 2007, to name a only few). Unfortunately, none of the earlier methods or tools could easily be adjusted to work with TRmorph output. For the present work, we have implemented a new morphological disambiguation system, that we briefly describe below.

Turkish morphological disambiguation is often viewed as a POS tagging with a few additional difficulties. These difficulties include (1) the data sparseness problems due to large tagset, (2) the difficulties of applying standard POS tagging algorithms because of variable number or inflectional groups in alternative analyses of a word and (3) the limited utility of features extracted from the local context of words in history-based POS tagging algorithms due to free-word-order nature of the language.

Like earlier work, we alleviate the data sparseness problem by making use of IGs. However, we do not view the morphological disambiguation in the usual setting of sequence labeling with hidden (POS) labels. We exploit the fact that the analyzer limits the choices for possible morphological analysis of a word, and the analyzer output is available in both training and testing time for the complete sentence. We extract a set of features, Φ, from all analyses offered by the morphological analyzer for the input sentence. Some features depend on the position of the word containing the IG, e.g., 'the last POS tag of the previous word', some features are word-internal, e.g., 'the word is capitalized', and others are general features extracted from the sentence or the available analyses of it, e.g., 'the analyses contain a finite verb'.

Recalling that an IG contains a root (or derivational morpheme) r, a POS tag c, and a set of inflections f, we assume that given Φ, analysis of a word is independent from the other words in the sentence, and similarly, an IG is independent of the other IGs in the same word given Φ. This allows us to define analysis of a word with m inflectional groups as,

$$\prod_{i=1}^{m} P(f|r,c,\Phi)P(r|c,\Phi)P(c|\Phi) \quad (1)$$

We estimate components of Equation 1 using discriminative models (logistic regression models for the results reported here). The resulting disambiguator has an accuracy of $91.2\,\%$ on the METU-Sabancı treebank with 10-fold cross validation. Although the results may not be directly comparable due to use of different morphological analyzers, this is similar to earlier results obtained on the same data set using Sak et al.'s (2007) disambiguator by Çetinoğlu (2014) ($90.2\,\%$ with a similar setting).

2.3 Dependency parsing

Since the syntactic relations in Turkish are between inflectional groups, rather than words, the dependency links relate IGs. A dependency parse of Example 1 is given in Figure 1.

For dependency parsing, we currently include an additional model to an already existing web service based on MaltParser (Nivre et al. 2006). The model is trained on the METU-Sabancı treebank. We use the version used in CoNNL-X shared task (Buchholz and Marsi 2006) with minor corrections. The resulting model has a labeled attachment score of 66.8 and unlabeled attachment score of 77.2 (10-fold-cross validation on the METU-Sabancı treebank). The results are obtained with coarse POS tags, with default learning method and without additional features or optimization. Initial experiments with additional features did not yield substantially better results. The (rather low) numbers we obtain are similar to earlier parsing results reported in the literature. Parsing Turkish was found to be difficult in earlier studies (Buchholz and Marsi 2006). Part of this difficulty seems to stem from the properties of the language, some of which are discussed above. However, our initial impression is that difficulty also stems from the small and not-very-high-quality resources available for the language. The only treebank available for Turkish (METU-Sabancı treebank) contains only 5 635 sentences and 56 424 tokens, and includes many annotation errrors and some unusual annotation choices.

3 Concluding remarks

We summarized the effort of integrating a Turkish NLP pipeline into the WebLicht infrastructure. The pipeline contains web services for sentence and word tokenization, morphological analysis and disambiguation, and dependency parsing. For some of the services, we used existing tools with some improvements and customization, and for others we developed some in-house tools. The tools and services described in this document are fully implemented and ready for use, and we are still improving some of the services. The services only make use of freely available tools, and the tools developed during this work will also be made available with a free/open-source license.

References

[Buchholz and Marsi 2006] Buchholz, Sabine and Erwin Marsi (2006). *CoNLL-X shared task on multilingual dependency parsing*. In: *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pp. 149–164.

[Çetinoğlu 2014] Çetinoğlu, Özlem (2014). *Turkish Treebank as a Gold Standard for Morphological Disambiguation and Its Influence on Parsing*. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Reykjavik, Iceland: European Language Resources Association (ELRA). ISBN: 978-2-9517408-8-4.

[Çöltekin 2010] Çöltekin, Çağrı (2010). *A freely available morphological analyzer for Turkish*. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*. Valetta, Malta, pp. 820–827.

[Çöltekin 2014] Çöltekin, Çağrı (2014). *A set of open source tools for Turkish natural language processing*. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Language Resources Association (ELRA).

[Eryiğit 2014] Eryiğit, Gülşen (2014). *ITU Turkish NLP Web Service*. In: *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 1–4.

[Hakkani-Tür et al. 2002] Hakkani-Tür, Dilek Z., Kemal Oflazer, and Gökhan Tür (2002). *Statistical Morphological Disambiguation for Agglutinative Languages*. In: *Computers and the Humanities* 36.4, pp. 381–410.

[E. Hinrichs et al. 2010] Hinrichs, Erhard, Marie Hinrichs, and Thomas Zastrow (2010). *WebLicht: Web-Based LRT Services for German*. In: *Proceedings of the ACL 2010 System Demonstrations*. Uppsala, Sweden: Association for Computational Linguistics, pp. 25–29.

[M. Hinrichs et al. 2010] Hinrichs, Marie, Thomas Zastrow, and Erhard Hinrichs (2010). *WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure*. In: *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. Ed. by N. Calzolari, K. Choukri, B. Maegaard,

J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias. Valletta, Malta: European Language Resources Association (ELRA). ISBN: 2-9517408-6-7.

[Nivre et al. 2006] Nivre, Joakim, Johan Hall, and Jens Nilsson (2006). *Maltparser: A data-driven parser-generator for dependency parsing*. In: *Proceedings of LREC*, pp. 2216–2219.

[Quasthoff et al. 2014] Quasthoff, Uwe, Dirk Goldhahn, and Thomas Eckart (2014). "Building Large Resources for Text Mining: The Leipzig Corpora Collection". In: *Text Mining*. Ed. by Chris Biemann and Alexander Mehler. Theory and Applications of Natural Language Processing. Springer, pp. 3–24. ISBN: 978-3-319-12654-8. DOI: 10.1007/978-3-319-12655-5_1.

[Sak et al. 2007] Sak, Haşim, Tunga Güngör, and Murat Saraçlar (2007). *Morphological Disambiguation of Turkish Text with Perceptron Algorithm*. In: *CICLing 2007*. Vol. LNCS 4394, pp. 107–118.

[Say et al. 2002] Say, Bilge, 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*. Eastern Mediterranean University, Cyprus.

[Yüret and Türe 2006] Yüret, Deniz and Ferhan Türe (2006). *Learning morphological disambiguation rules for Turkish*. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. HLT-NAACL '06. New York, pp. 328–334. DOI: 10.3115/1220835.1220877.