The Annotation Process of the ITU Web Treebank

Tuğba Pamay, Umut Sulubacak, Dilara Torunoğlu-Selamet, Gülşen Eryiğit
Department of Computer Engineering
Istanbul Technical University
Istanbul, 34469, Turkey
{pamay,sulubacak,torunoglud,gulsen.cebiroglu}@itu.edu.tr

Abstract

The potential of processing user-generated texts freely available on the web is widely recognized, but due to the non-canonical nature of the language used in the web, it is not possible to process these data using conventional methodologies designed for well-edited formal texts. Procedures for properly annotating raw web data have not been as extensively researched as those for annotating well-edited texts, as also evident from the viewpoint of Turkish language processing. Moreover, there is a considerable shortage of human-annotated corpora derived from Turkish web data. The ITU Web Treebank is the first attempt for a diverse corpus compiled from Turkish texts found on the web. In this paper, we first present our survey of the non-canonical aspects of the language used in the Turkish web. Next, we discuss in detail the annotation procedure followed in the ITU Web Treebank, revised for compatibility with the language of the web. Finally, we describe the web-based annotation tool following this procedure, on which the treebank was annotated.

1 Introduction

As researchers grow more conscious of the potential of applications on user-generated web data, developing methodologies for processing the language of the web becomes increasingly important. The amount of raw data freely available on the web is not only massive, but also it is constantly being expanded and renewed. As such, if web data were to be processed as accurately as edited texts which have been in the spotlight for a long time, they would constitute a data source substantially larger than any human-annotated corpus to date, bolstering up research on unsupervised and semi-supervised learning.

Despite the potential, processing web data is a challenge for any system designed for or trained on edited texts, due to radical differences in the languages employed in the domains. The Internet has its own idiosyncratic language that is very loose and colloquial compared to the formal language standard. Web users are often not concerned with grammar and directly transcribe their spontaneous speech to their writing. The language of the Internet is also highly memetic and dominated by various sub-cultures. Often, users experiment with their own house rules instead of canonical grammar, omitting letters or replacing them with foreign characters, deliberately making spelling mistakes and putting words in inappropriate letter cases. Such practices render the language of the Internet highly non-canonical and complicate the processing of web data.

The ITU Web Treebank is a data set containing sentences collected from various domains on the Internet, inspired by recent efforts on other languages (Seddah et al., 2012; Bies et al., 2012). In the absence of Turkish language resources originating from the web, the ITU Web Treebank aimed to establish the first manually annotated web language resource for Turkish. Sulubacak and Eryiğit (2014) described the annotation procedure of the ITU Web Treebank in detail, outlining the treebank composition, the annotation setting and the syntactic framework. Another aim of the ITU Web Treebank was to put forward and demonstrate an approach for annotating the non-canonical language found in the web. This paper goes into detail and thoroughly describes

Proceedings of LAW IX - The 9th Linguistic Annotation Workshop, pages 95–101,
Denver, Colorado, June 5, 2015.

this approach, along with the motivations for any changes proposed over the previous de facto annotation standard for Turkish.

Section 2 discusses the non-canonical elements commonly found in the language of the web. Section 3 goes into detail about our annotation procedure and the layered structure of the ITU Web Treebank. Finally, Section 4 introduces the updated annotation tool we used in annotating the ITU Web Treebank.

2 Non-Canonical Forms in Web

The language of the web is not entirely arbitrary, and it is still possible to work out the ways in which it differs from canonical language. The colloquial expressions and peculiar grammatical conventions still reveal a pattern, and word usages can be likened to an elaborate jargon. We present our analysis of the non-canonical aspects of the language of the Turkish web below, in an exclusive category for each aspect.

Punctuation: Punctuation is very often omitted by users on the web, especially in daily conversations. Especially abundant in social media, where posts are usually directed to each user's own limited network rather than the public, this phenomenon is not limited to terminal periods and also often affects punctuation like commas, semicolons and apostrophes that serve as constituent or morpheme boundaries. The omission of terminal punctuation overly complicate the task of splitting sentences that are semantically independent but syntactically appear as a single sentence. Moreover, syntactically similar constituents pose a challenge for syntactic annotation and parsing when they are not properly disjoined by punctuation.

Abbreviated Writing: Whether forced by websites such as microblogs that impose a character limit on messages or motivated by a need to respond quickly to the dynamics of a social medium, there is a widespread trend of using abbreviations and shorthand on the web. As abbreviated writing is manifested in a variety of ways, it is a major challenge to handle such expressions.

Exaggerated Writing: Another spelling anomaly is manifested in excessively repeated letters, usually vowels, in order to emphasize an expression or convey frustration, excitement or exclamation. These expressions often correspond to interjections and other vocatives.

Spelling Mistakes: Mistakes in spelling are among the most commonly occurring aspects in informal language, and they can be encountered in virtually any platform on the web. While some spelling errors can be made deliberately as part of a jargon, they most commonly stem from overlooked typing mistakes, as it is not common practice to double-check typing.

Foreign Characters: Internet users may prefer not to use the letters in the original spelling a word for a variety of reasons. For instance, non-letter characters may be substituted for regular letters with similar shapes, in order to adapt to experimental spelling trends. Also, because some platforms restricting character encoding do not support certain Turkish letters such as 'ç', 'ğ', 'ş', 'ı', 'İ', 'ö' and 'ü', users may be forced to use the closest ASCII versions 'c', 'g', 's', 'i', 'I', 'o' and 'u'. Moreover, certain input methods may not provide a convenient means to type non-English letters, further encouraging users to make the ASCII substitution.

Letter Case: A significant portion of the users on the web do not attach importance to letter cases. Capitalizing the initial letters of proper nouns and the first words of sentences is often disregarded, abbreviations in uppercase are occasionally typed out in lowercase, and stylizing certain proper nouns in mixed case is also frequently neglected.

Web Entities: It has become fairly common for web users to share URLs and e-mail addresses with other users from their private networks. Additionally, with the advent of Twitter, microblogging services call for the active usage of mentions, hashtags and other metadata tags. The usage of emoticons to express feelings in plain text has also become extremely popular. As such web-specific entities may exhibit irregular morphology and syntax, it is necessary to detect and handle them.

3 Annotation Procedure

In order to attain a more proper and lenient framework for annotating non-canonical language, we revised the entire annotation procedure since Atalay et al. (2003) and made several extensions and modifications. Going by the common convention, we processed our raw data through consecutive steps, each establishing a separate aspect of the data. Before any morphological or syntactic tagging, we applied an extensive normalization routine in order to facilitate the processing of the data by the later modules. We also updated our morphological and syntactic annotation schemes and designated particular morphological tags and dependency labels in an attempt to formalize various morphosyntactic phenomena common in the language of the web.

The ITU Web Treebank is organized in three cascading layers: 1) The normalization layer, 2) The morphology layer and 3) The syntax layer. Annotating data involves firstly the manual normalization, and then the consecutive morphological and dependency tagging of the data. Starting from the raw data, the result of each annotation phase contributes to the next layer of the treebank.

The cascading nature of the layers on the raw data makes it possible to compare each successive layer and extract training and validation corpora from the data. As such, the ITU Web Treebank comprises major resources for both training and validating systems aiming to automatize tasks corresponding to each annotation phase, such as automatic normalizers, morphological disambiguators and dependency parsers, each naturally attuned to non-canonical web data.

The subsections below provide a description of our annotation phases and the changes made on each phase in adapting to the non-canonical language of the Internet.

3.1 Normalization

Our manual normalization phase acts as a preprocessing routine before morphological annotation. Because morphological and syntactic taggers are typically designed to process formal language and would require a radical redesign to handle non-canonical language on their own, normalization is called for as an initial step also in automatically processing non-canonical language. In this phase, we manually tokenize raw sentences and process each token in order to eliminate any errors in spelling and word cases, expand non-standard abbreviations and contractions, and mark web entities such as URLs for later phases, as established in Eryiğit and Torunoğlu-Selamet (2015).

We investigate the following issues during manual normalization.

Abbreviations: We replace informal abbreviations such as ***kib*** for ***kendine iyi bak*** ("*take care of yourself*") with their full forms. Institutionalized and formal abbreviations used for entire classes of words such as titles like ***dr*** for ***doktor*** ("*doctor*") and units of measurement like ***kg*** for ***kilogram*** are left as they are, to be handled later in the morphology layer.

Shorthand: We fully type out shorthand that omits or replaces certain characters and leaves out a fragment from which it is still possible to guess the full form. Such usages may omit any non-initial vowel as in ***anldm*** for ***anladım*** ("*I get it*"), the postvocalic 'ğ' as in ***saol*** for ***sağ ol*** ("*thanks*"), and other elided consonants such as a postconsonantal 'h' as in ***mrb*** for ***merhaba*** ("*hello*"). Shorthand may also involve contractions such as ***naber*** for ***ne haber*** ("*what's up*"), as well as heavily assimilated or broken verb suffixes typed out as though they were pronounced with a nonstandard accent as in ***-yon*** for ***-yorsun*** (the present progessive tense, 2nd person suffix).

Web Entities: We enclose all URLs, e-mail addresses, mentions, hashtags and emoticons in corresponding tags for each class, respectively `@url[...]`, `@email[...]`, `@mention[...]`, `@hashtag[...]` and `@smiley[...]`. These classes of web-specific tokens are often found to deviate from regular punctuation (for emoticons) and nouns (for the rest of the classes) in their participation in syntax. By applying these tags, we provide clues to the morphological analyzer so that it would generate special morphological features for these semantic classes of tokens, which in turn provide clues to the syntactic parser.

Letter Case: We investigate the letter cases of each token and make corrections as necessary. This is among the most demanded tasks, since the capitalization of sentence-initial tokens and proper nouns are very commonly omitted in the language of the web. The task is however not limited to capitalization, as it is sometimes proper to put tokens in uppercase (e.g. in *"NAACL"*) or even mixed case (e.g. in *"LaTeX"*), as well as decapitalize tokens that should have been in lowercase. This task is also quite important, since morphological analyzers are often case-sensitive and may not work properly with inputs in wrong letter cases.

Character Repetition: We eliminate excessive character repetitions, excluding punctuation, often used for exclamation or emphasis as in ***lütfeeeeen*** ("*pleeeeease*").

Improper Glyphs: We restore the appropriate Turkish letters whenever they are replaced by a non-Turkish letter or a non-letter character as in ***$aqa*** instead of ***şaka*** ("*joke*"). This is roughly equivalent to the *Leetspeak* of the English web, practiced to add some humorous flair to the language, though rather uncommonly. A more common practice is to use the closest ASCII versions of non-English letters in the Turkish alphabet as in ***cus*** for ***çüş*** ("*whoa*"), and replacing such letters is also part of this task.

Spelling Mistakes: As should be intuitive, we also correct any remaining spelling mistakes after all the aforementioned checks are completed.

3.2 Morphology

The next phase after normalization is the morphological tagging phase. Since morphological analyzers would be able to automatically process the data after normalization, the phase usually amounts to manually disambiguating between automatically generated morphological analyses for each token. We use a version of the morphological framework described in Şahin et al. (2013), with some additional fine POS categories integrated in order to properly annotate certain elements of non-canonical language. For such, it is also occasionally required to manually provide morphological analyses when a token is not analyzed properly by the base analyzer due to its non-canonical aspects.

One of our significant additions to the framework is the support for formally acceptable abbreviations, which are automatically assigned their full forms as their lemmata and treated as nouns with the newly introduced fine POS `+Abbr`, such as units of measurement. Not only does this increase the expressiveness of the framework for formal texts, but also it takes a significant burden from the normalization phase by removing the need to replace most abbreviations commonly used on the web with their full forms. However, as discussed before in Section 3.1, certain abbreviations representing multiple words and other non-standard abbreviations do not fall under this scope.

Our other major revision involves the morphological annotation of web entities, as outlined previously in Section 3.1. Such entities often have idiosyncratic usages deviating from those of regular tokens with the same assigned POS, and parsers therefore require an alternative cue in order to distinguish these entities and learn the exclusive syntax applying to them. In our framework, emoticons are unambiguously treated like punctuation, and this is reflected in their morphological features by tagging them as punctuation with the fine POS `+Smiley`. Other web entities are treated as nouns in the same manner, with the fine POS `+URL`, `+Email`, `+Mention` and `+Hashtag`. For a different viewpoint, Foster et al. (2011) automatically assign generic surface forms like *Username* and *Hashtag* to such web entities, letting the parser discern them by their lexical features. However, we find that encoding this information in morphology as in Gimpel et al. (2011) allows our data-driven parsers to successfully distinguish these entities without obscuring their original lexical features. We facilitate the morphological tagging of web entities with the help of a pre-tagger processing the lexical tags assigned in the normalization phase, as explained previously in Section 3.1.

3.3 Syntax

The third and last phase of annotation is the dependency parsing of the normalized and morphologically tagged tokens. We follow the revised, web-compatible dependency annotation framework de-

scribed in Sulubacak and Eryiğit (2014), which also introduced the ITU Web Treebank for the first time. This framework considers many aspects of the non-canonical language of the web and offers comprehensive and convenient annotation schemes to express them.

Our updated annotation scheme takes care not to make any assumptions about the syntactic structures of sentences outside of the most fundamental elements. Dependencies to tokens that may be left out in sentences found on the web are eliminated whenever possible. The annotation schemes of coordination and relativizer structures, sentence predicates and punctuation are all revised as part of this effort. Additionally, certain restrictions on the root node are relaxed, so that multiple constituents can now depend on the root node, even though the root node itself is not allowed to have a head. Constituents depending on the root node can also be assigned any permissible dependency relation rather than the single dummy relation **ROOT**, allowing for more semantically appropriate annotation schemes for constituents like predicates and vocatives that essentially modify the sentence. The full set of changes on the dependency grammar are described in Sulubacak and Eryiğit (2014).

4 Annotation Tool

In this study, we introduce an updated version of the ITU Treebank Annotation Tool (Eryiğit, 2007) to annotate the ITU Web Treebank. The new version is a web-based application supporting annotation for the normalization layer in addition to the morphology and syntax layers, allowing concurrent operation by multiple annotators on the same data.

The new version of the annotation tool comes with a set of changes in the annotation interfaces in compliance with the changes in the annotation methodologies for web data compatibility. The tool can now automatically generate morphological analyses for certain orthographically tagged tokens such as web entities in addition to the output fetched from a morphological analyzer, to be later disambiguated by hand. The dependency annotation interface now supports the specification of multiple head tokens for a given constituent, allowing the annotation of deep dependencies on the tool while still enforcing at least one head for each dependent. The interface also displays the root node as a separate token and allows regular dependencies to the root node.

In addition to the annotation of the ITU Web Treebank, our updated annotation tool is used in the creation of the revised IVS (Eryiğit and Pamay, 2007 2014) Corpus and the IMST (Sulubacak and Eryiğit, 2014) Corpus, as well as the validation corpus for the Turkish mobile assistant developed by Çelikkaya and Eryiğit (2014).

The annotation interfaces of our new tool are shown in Figures 1 and 2. Figure 1 displays the normalization and morphological tagging interfaces on a unified window, whereas Figure 2 shows the syntactic tagging interface along with the dependency relation table for the sentence being processed.

5 Conclusion

In this paper, we presented the web-compatible revision of our annotation procedure, which we used to annotate the ITU Web Treebank, the first manually annotated web treebank for Turkish organized in three layers, namely normalization, morphology and syntax. We provided a survey of new expressions common in the non-canonical language of the web, and detailed the measures we took in order to handle them during normalization, morphological tagging and dependency annotation. We described the new version of our treebank annotation tool updated in accordance with these measures. We believe the layered annotation framework we outlined in this work would serve as an effective baseline for any study involving the annotation of non-canonical web data.

Acknowledgments

We would like to acknowledge that this work is part of a research project entitled "Parsing Web 2.0 Sentences" subsidized by the TÜBİTAK (Turkish Scientific and Technological Research Council) 1001 program (grant number 112E276) and part of the ICT COST Action IC1207. We would hereby like to offer our sincere gratitude to our colleagues Ayşenur Genç, Can Özbey, Kübra Adalı and Gözde Gül Şahin who offered additional help during the annotation phase.

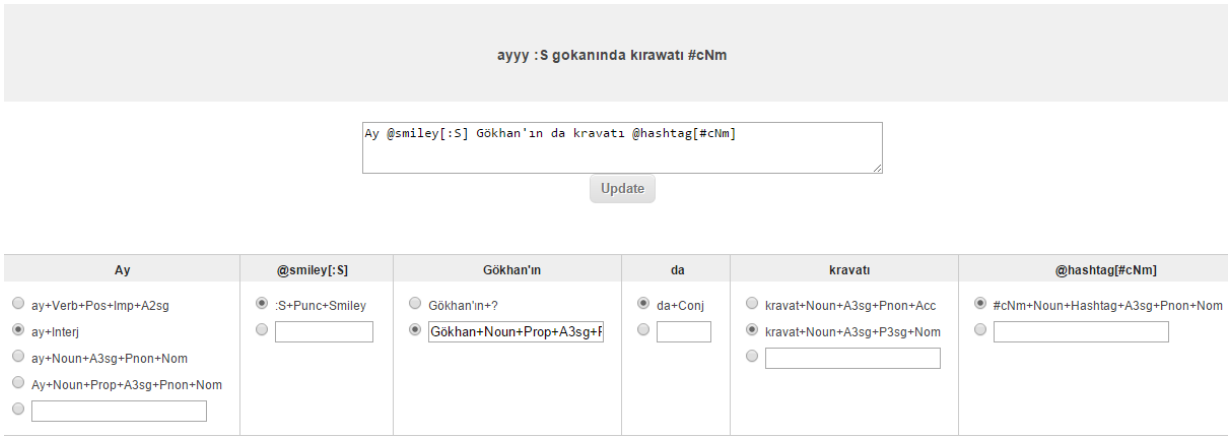

Figure 1: **The Normalization and Morphological Tagging Phases**
A snapshot from the unified normalization and morphological tagging screen from the new annotation tool. The example shows the hypothetical Turkish tweet "***ayyy :S gokanında kırawatı #cNm***", roughly translated to English as "*ohhh :S gokans tie too #aWw*", normalized as "***Ay @smiley[:S] Gökhan'ın da kravatı @hashtag[#cNm]***". The morphology window displays three different cases for tokens where the annotator 1) manually disambiguates between generated morphological analyses, 2) verifies a morphological analysis automatically derived from orthographic tags, or 3) has to manually type in an analysis.

Relation Table

Ay	Root	VOCATIVE	0	Delete
@smiley[:S]	Ay	PUNCTUATION	0	Delete
Gökhan'ın	kravatı	POSSESSOR	0	Delete
da	Gökhan'ın	CONJUNCTION	0	Delete
kravatı	Root	PREDICATE	0	Delete
@hashtag[#cNm]	Root	VOCATIVE	0	Delete

Figure 2: **The Syntactic Annotation Phases**
A snapshot showing the syntactic annotation screen of the new annotation tool. The example shows the normalized and morphologically tagged sentence marked for dependencies. Each row of the relation table corresponds to a dependency arc, where the columns respectively denote the dependent token, the head token, the dependency relation, and the inflectional group index of the head token.

References

Nart B Atalay, Kemal Oflazer, Bilge Say, et al. 2003. The annotation process in the Turkish treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreteted Corpora (LINC)*.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. *Linguistic Data Consortium*.

Gökhan Çelikkaya and Gülşen Eryiğit. 2014. A mobile assistant for Turkish. In *Proceedings of the 2nd International Conference on Turkic Language Processing (TURKLANG)*, Istanbul, 06-07 November.

Gülşen Eryiğit and Tuğba Pamay. 2007-2014. ITU Validation Set for Metu-Sabancı Turkish treebank. In *Proceedings of the 2nd International Conference on Turkic Language Processing (TURKLANG)*, Istanbul, 06-07 November.

Gülşen Eryiğit and Dilara Torunoğlu-Selamet. 2015. Social media text normalization for Turkish. (Under review).

Gülşen Eryiğit. 2007. ITU treebank annotation tool. In *Proceedings of the 1st Linguistic Annotation Workshop (LAW)*, pages 117–120. Association for Computational Linguistics.

Jennifer Foster, Özlem Çetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. #hardtoparse: POS tagging and parsing the Twitterverse. In *AAAI 2011 Workshop on Analyzing Microtext*, pages 20–25.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47. Association for Computational Linguistics.

Muhammet Şahin, Umut Sulubacak, and Gülşen Eryiğit. 2013. Redefinition of Turkish morphology using flag diacritics. In *Proceedings of The 10th Symposium on Natural Language Processing (SNLP-2013), Phuket, Thailand, October*.

Djamé Seddah, Benoit Sagot, Marie Candito, Virginie Mouilleron, and Vanessa Combet. 2012. The French Social Media Bank: a treebank of noisy user generated content. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*.

Umut Sulubacak and Gülşen Eryiğit. 2014. A redefined Turkish dependency grammar and its implementations: A new Turkish web treebank & the revised Turkish treebank. (Under review).