Parsing Tweets into Universal Dependencies

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
`email@domain`

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
`email@domain`

Abstract

1 Introduction

Analyzing the syntax of tweets is challenging for traditional NLP tools because most of the tweet texts are informal and noisy.

In this paper, we propose to parse the tweets in the convention of universal dependencies and built up the whole pipeline to parse the tweets from the raw text form.

Contribution of this paper includes:

- We create a new version of tweet Treebank Tweebank 2.0
- We propose a neural network method to parse tweets into universal dependencies
- We study the adaptation of universal dependencies for analyzing tweets

2 Related Work

Eisenstein (2013) reviewed NLP approaches for analyzing text on social media, especially for tweets and showed that there are two major directions for NLP community to handle the tweets, including normalization and domain adaptation. He also pointed out that normalization can be problematic because precisely defining the normalization task is difficult.

Kong et al. (2014) argues that the Penn Treebank approach to annotation is poorly suited to more informal genres of text, as some of the annotation challenges for tweets, including token selection, multiword expressions, multiple roots, and structure within noun phrases diverge significantly from conventional approaches. They believe that rapid, small scale annotation efforts performed by imperfectly-trained annotators should provide enough evidence to train an effective parser, given the rapidly changing nature of tweets (Eisenstein, 2013), the attested difficulties of domain adaptation for parsing (Dredze et al., 2007), and the expense of creating Penn Treebank-style annotations (Marcus et al., 1993). Therefore, they build a new corpus of tweets (Tweebank), with conventions informed by the domain, using new syntactic annotations that can tackle all the forementioned problems annotated in a day by two dozen annotators, most of whom had only cursory training in the annotation scheme. Then, they modify the decoder of the TurboParser, a graph-based dependency parser, which is open-source and has been found to perform well on a range of parsing problems in different languages (Martins et al., 2013) to adapt to the Tweebank dataset, and incorporate new features such as Brown Clusters and Penn Treebank features and changes to specification in the output space into TurboParser.

3 Data Annotation

4 Pipeline

4.1 Tokenization

We use the UDPipe[1] to tokenize the tweets and then detokenize the wrongly tokenized usernames and hashtags.

[1] https://github.com/ufal/udpipe

4.2 POS Tagging

Different from standard sentences, special constructions in tweets exist in both the whole tweet-level and token-level.

4.2.1 Token-level Unconventional Part-of-Speech

In addition to standard language, tweets have the following specific constructions:

- Discourse marker, URL, hashtag, emoticon
  - RT, http://bit.ly/xyz, #acl, :-), @user
- Abbreviations
  - wtf (what the fuck), smh (shake my hand), mfw (my face when), ima (i am going to), rn (right now), af (as fuck)
- Contraction
  - Contraction with apostrophe: Theres, hes, bookll, buyem
  - Contraction without apostrophe: gonna, tryna, gimme, Iv, im, aint, whatis

Gimpel et al. (2011) proposed a set of part-of-speech tags that handles most of these cases. We are inspired by their work but argue that, first different abbreviation can have different syntactic functions. Like mfw is usually followed by an adverbial clause and ima is usually followed by a clausal complement. It is not reasonable to treat them in the same part-of-speech. In this paper, when annotating the POS tagging for abbreviations, we first try to recover their original forms, then use the POS of the core-word as the POS for the abbreviation. Second, four special POS tags (S, L, M, Y) were designed to handle contraction words in Gimpel et al. (2011). Major concern of designing such tags is to minimize the effort of tokenization. However, contractions of common nouns and pronouns are casted into the same category which increase the difficulty of distinguishing their syntactic function (say, there's and book'll are treated with the same syntactic function). What's more, only a small proportion of words can be categorized into these tags (2.7 % in total), which cast a doubt of the usefulness of these certain tags. In this paper, we believe such contraction can be properly handled by tokenization module, so we suggest to tokenize the contraction word and annotate POS tag accordingly. Besides the contraction that be conventionally tokenized, tweets also witness a set of unconventional contraction like iv (I've), whatis (what is). In this paper, we follow the same idea of annotation abbreviation to handle the unconventional contractions and use the POS of core word of the original form as their POS. Third, special POS was designed to handle emoticon in Gimpel et al. (2011). However, in most cases, emoticon plays the same role as most of the symbolic tokens. In this paper, we follow the UD guideline to annotate the emoticon as symbol (SYM). At last, its arguable that some of the hashtags, URLs can work as a nominal in tweets. Whether treating them as the same part-of-speech or different ones according to their context is an open question. A preliminary survey on the standard UD English data shows that URL, email address are all tagged as the foreign language (X), so we also tag them as X and leave the disambiguation of their syntactic function to the annotation of parse tree.

4.2.2 Tweets-level Special Construction

There are several tweet-level constructions which are unconventional to standard text, including:

- Retweet: RT @user : ⟨ sentence ⟩
- Leading or ending topic marked as hashtag: #topic #topic #topic ⟨ sentence ⟩ #topic #topic #topic
- Leading or ending complementary URL: ⟨ complementary URL ⟩ ⟨ sentence ⟩ ⟨ complementary URL ⟩

4.3 Sentence Segmentation

5 Model

6 Experiments

7 Conclusion

References

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1051–1055, Prague, Czech Republic, June. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia, June. Association for Computational Linguistics.

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: Short Papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012, Doha, Qatar, October. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622, Sofia, Bulgaria, August. Association for Computational Linguistics.