The Denoised Web Treebank: Evaluating Dependency Parsing under Noisy Input Conditions

Joachim Daiber, Rob van der Goot
ILLC University of Amsterdam, CLCG University of Groningen
J.Daiber@uva.nl, R.van.der.Goot@rug.nl

Abstract

We introduce the Denoised Web Treebank: a treebank including a normalization layer and a corresponding evaluation metric for dependency parsing of noisy text, such as Tweets. This benchmark enables the evaluation of parser robustness as well as text normalization methods, including normalization as machine translation and unsupervised lexical normalization, directly on syntactic trees. Experiments show that text normalization together with a combination of domain-specific and generic part-of-speech taggers can lead to a significant improvement in parsing accuracy on this test set.

Keywords: Parsing, Part-of-Speech Tagging, Social Media Processing, Web Treebank

1. Introduction

The quality of automatic syntactic analysis of clean, in-domain text has improved steadily in recent decades. Out-of-domain text and grammatically noisy text, on the other hand, remain an obstacle and often lead to significant decreases in parsing accuracy. Recently, a lot of effort has been put into adapting natural language processing tools, such as named entity recognition (Liu et al., 2012) and POS tagging (Gimpel et al., 2011), to noisy content. In this paper, we focus on dependency parsing of noisy text. Specifically, we are interested in how much parse quality can be gained by text normalization. For this, we introduce a new dependency treebank with a normalization layer. This new dataset can be used to quantify the influence of text normalization on the parsing of user-generated content.

The contributions of this paper are as follows: (1) We introduce the Denoised Web Treebank, a new Twitter dependency treebank with a normalization layer; (2) we propose a corresponding noise-aware evaluation metric; and (3) we use this dataset and the metric as a benchmark to evaluate the impact of text normalization on dependency parsing of user-generated content.

2. Related Work

For the domain of web data, various datasets and treebanks have been introduced. Table 1 provides an overview of all relevant English treebanks.

The constituency treebanks mentioned here were created using the English Web Treebank annotation guidelines (Bies et al., 2012), which are an addendum to the Penn Treebank guidelines (Bies et al., 1995). These guidelines discuss domain-specific phenomena, including adaptations of existing labels as well as the addition of new labels for novel linguistic constructions. Foster et al. (2011a) describe a constituency treebank consisting of two domains; Twitter and sports forums. The Twitter part is of comparable size to our treebank and is described in more detail in Foster et al. (2011b).

The dependency treebanks show greater diversity in annotation. The English Web Treebank (Silveira et al., 2014) is annotated using the Universal Dependencies guidelines with additional relation types for the web domain. A very different approach is taken for the annotation of the Tweebank (Kong et al., 2014). In its format, individual words can be skipped in the annotation. This is motivated by the idea that not all words in a Tweet contribute significantly to the syntactic structure and their inclusion would lead to arbitrary decisions unhelpful for most downstream applications. Additionally, because Tweets are used as units instead of sentences, having multiple roots is allowed. This adjusted dependency format makes it harder to use existing parsers with this dataset.

The Foreebank (Kaljahi et al., 2015), a treebank focusing on forum text, is the only other treebank that includes normalization annotation. It includes manual normalizations of the raw text, and constituency trees of the normalized sentences. The normalization is kept as minimal as possible and is represented in the tree by appending an error suffix to the POS tags. The Foreebank allows analysis of the effect of different errors on the parsing performance of a constituency parser.

Our contribution, the Denoised Web Treebank, fills the gap of a native (i.e., non-converted) dependency treebank including normalizations for the web domain. In the past, automatic conversions were used for this task (Petrov and McDonald, 2012; Foster et al., 2011a) using the Stanford Converter (De Marneffe et al., 2006). But for the noisy web domain, the conversions might be of questionable quality.

Previous work on the parsing of web data has mostly focused on complementing existing parsers with semi-supervised data The amount of training data can be artificially enlarged by using self-training or up-training (Petrov and McDonald, 2012; Foster et al., 2011b). Another source of semi-supervised improvements can be gained from using features gathered from large amounts of unannotated texts (Kong et al., 2014). A completely different approach is taken by Khan et al. (2013), where the most appropriate training trees are found in the train treebank for each sentence.

3. Dataset

3.1. Data Preparation

We collected all Tweets within a period of 24 hours from January 07, 2012 00:00 until 23:59 GMT. To avoid possible

Name	Number of trees	OOV[1]	Average sent. length	Annotation style	Normal-ization	Source
English Web Treebank	16,622	28%	16.4	Constituency and Dependency	No	Yahoo! answers, e-mails, newsgroups, reviews, blogs
Foster et al. (2011a)	1,000	25%	14.0	Constituency	No	Twitter, sports forums
Foreebank	1,000	29%	15.6	Constituency	Yes	Technical forums
Tweebank	929	48%	13.3	Dependency	No	Twitter
Denoised Web Treebank	500	31%	10.4	Dependency	Yes	Twitter

Table 1: English treebanks based on user-generated content.

biases of automatic language identification tools towards well-formed language, we manually classified the Tweets in random order into English and non-English Tweets until we reached a reasonably-sized corpus of Tweets classified as English. We then manually split this corpus into sentences and randomly selected 250 sentences as a development set and 250 sentences as a test set. Table 1 compares some basic statistics of this treebank against other Web treebanks. Out-of-vocabulary rate is calculated against the English dictionary of the GNU Aspell spell checker.[1]

3.2. Normalization

The goal of the normalization was to leave the original tokens intact and not to replace them by their normalized forms directly. Hence, we keep both the original tokens and the normalized version of the sentences with word alignments. Figure 1 depicts a gold standard dependency graph including the alignments to the original tokens.

Abbreviations Abbreviations and slang expressions are expanded whenever necessary for syntactic reasons. Examples include instances such as "cu", used as the short form of *see* and *you*, which as a single token would include both the verb and the object of the sentence.

Punctuation Punctuation is inserted if it is necessary to disambiguate the sentence meaning. Emoticons, such as `:)`, are kept intact.

Zero copulas The data contains several cases of zero copula, i.e. a copula verb is not realized in the sentence. These occurrences are annotated by inserting the copula verb in the normalized version of the sentence (see Figure 1).

3.3. Syntactic Annotation

The normalized tokens were automatically parsed using a generative phrase structure parser[2] and then converted to dependencies. Both part-of-speech tags and dependency annotations were then manually corrected in two passes. The dependency annotations follow the format of the CoNLL-X shared task on multilingual dependency parsing (Buchholz and Marsi, 2006), with the following adaptations:

Emoticons Emoticons are kept intact, tagged with the part-of-speech tag `UH` (interjection) and are attached to the head of the sentence.

[1]English Aspell dictionary: `http://aspell.net/`

[2]`https://code.google.com/p/berkeleyparser/`

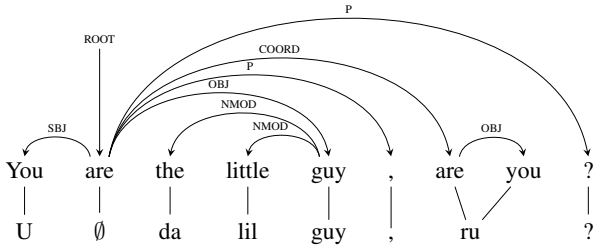

Figure 1: Aligned dependency graph.

Domain-specific elements Twitter-specific syntax was treated as follows: Usernames at the start of a sentence that do not fulfill a syntactic role, e.g. as the subject, are attached to the main verb using the `DEP` (unclassified) dependency relation. In all other cases, usernames are treated as proper nouns. *RT* and similar markers, as well as hashtags indicating the topic of a Tweet (e.g. `#worldcup`) are attached to the main verb as `DEP`.

4. Evaluating Noise-Aware Parsing

Our dataset provides alignments between the gold standard and the original tokens, allowing for insertions, deletions and modifications. Hence, the standard dependency parsing metrics, unlabeled and labeled attachment scores, are no longer sufficient. In our dataset, there may not be a direct one-to-one correspondence between the predicted tree and the gold tree. Hence, we allow the parser to make any insertions, deletions and modifications to the tokens under the assumption that it provides an alignment between the modified tokens and the original tokens. The evaluation is then performed using a metric based on precision and recall values calculated using these alignments.

Aligned precision and recall

Based on the normalized side of the gold standard and the parser's aligned predictions, we calculate precision, recall and F_1 score for dependencies (Eq. 1–3). We base the evaluation metric on the standard definitions of precision and recall, which are widely used in natural language processing. In Eq. 1 and 2, TP, FP and FN are the numbers of true positive, false positive, and false negative results. The F_1 measure is the harmonic mean of precision and recall.

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = \frac{TP}{TP + FN} \quad (2)$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3)$$

Definition Formally, when comparing the parser output against the dataset, the following information is provided for each instance:

- the original sentence S_O
- a predicted dependency tree $D_P = \langle V_P, E_P \rangle$
- a gold dependency tree $D_G = \langle V_G, E_G \rangle$
- alignment function a_P for predicted tokens
- alignment function a_G for gold tokens

For each parsed dependency tree, S_O is the sequence of original, non-normalized tokens. The two alignment functions a_G and a_P map the gold tokens and the predicted tokens to the original tokens in S_O. In the case of an insertion, the new token cannot be aligned to any of the original tokens in S_O and, therefore, such insertions are mapped to an artificial `NULL` token.

Unlabeled dependencies Based on all test instances, we calculate the total number of true positive, false positive and false negative dependency relations as follows: For each gold dependency tree $D_G = \langle V_G, E_G \rangle$ and each predicted dependency tree $D_P = \langle V_P, E_P \rangle$, let M_G and M_P be the set of dependency relations mapped to the original tokens in S_O:

$$M_G = \{\langle a_G(w_i), a_G(w_j)\rangle \mid \langle w_i, r, w_j\rangle \in E_G\}$$
$$M_P = \{\langle a_P(w_i), a_P(w_j)\rangle \mid \langle w_i, r, w_j\rangle \in E_P\}$$

The true positive, false positive and false negative dependency relations can then be calculated as:

$$TP = \sum_{\langle S_O, D_P, D_G, a_P, a_G\rangle} |M_G \cap M_P|$$
$$FP = \sum_{\langle S_O, D_P, D_G, a_P, a_G\rangle} |M_P \setminus M_G|$$
$$FN = \sum_{\langle S_O, D_P, D_G, a_P, a_G\rangle} |M_G \setminus M_P|$$

Labeled dependencies To measure labeled dependencies, the dependency type is added to the head-modifier pair in M_P and M_G:

$$M'_G = \{\langle a_G(w_i), r, a_G(w_j)\rangle \mid \langle w_i, r, w_j\rangle \in E_G\}$$
$$M'_P = \{\langle a_P(w_i), r, a_P(w_j)\rangle \mid \langle w_i, r, w_j\rangle \in E_P\}$$

Relation to other metrics This metric can be seen as a generalization of the commonly used attachment score measure. If there is a one-to-one alignment between every predicted token and every gold token, the unlabeled and labeled aligned F_1 scores are equivalent to the unlabeled (UAS) and labeled attachment score (LAS).

5. Experiments

Having introduced our dataset and the corresponding evaluation metric, we can evaluate the impact of two methods commonly used to aid in the parsing of noisy content: noise-robust part-of-speech tagging and text normalization.

5.1. Part-of-Speech Tagging

POS tagging is a necessary preprocessing step for many parsing algorithms. Previous studies (e.g., Foster et al. (2011b)) have shown that the accuracy of POS tagging can suffer significantly from noisy content. However, it is possible to adapt POS taggers to this type of input. In this experiment, we will briefly introduce approaches to adapting POS taggers and perform an evaluation on our dataset.

Domain-specific tagging Gimpel et al. (2011) present a domain-specific conditional random field POS tagger using a coarse part-of-speech tagset of 25 tags that was specifically designed for and trained on Twitter data. The tagset includes tags for social media-specific tokens, such as URLs, email addresses, emoticons, Twitter hashtags and usernames.

Role of POS tags in the parser For our experiments, we use the discriminative graph-based maximum spanning tree (MST) parser (McDonald et al., 2005). This dependency parser expects both fine- and coarse-grained tags as features in its well-established standard setting. Since we are interested in the influence of POS tagging on parse quality instead of the impact of individual features in the parser, we use this standard setting but combine the coarse-grained tags from the domain-specific tagger with the POS tags produced by a POS tagger with a less coarse-grained tagset. Both are combined by first determining n-best fine-grained tags for each token. For hidden Markov models, the probability of the tags occurring at a given position can be calculated using the forward-backward algorithm as $P(t_i = t) = \alpha_i(t)\beta_i(t)$, where $\alpha_i(t)$ is the total probability of all possible tag sequences ending in the tag t at the ith token and $\beta_i(t)$ is the total probability of all tag sequences starting from tag t at the ith token and continuing to the end of the sentence (Jurafsky and Martin, 2000; Prins, 2005). The n-best fine-grained tags are then combined with the coarse tags by a simple voting rule. Our experiments use a standard trigram HMM tagger[3] (Brants, 2000) and the OpenNLP maximum entropy tagger.[4]

Impact on parse quality Table 2 shows the influence of POS tagging on the performance of the MST parser on the development part of our dataset. Statistical significance testing is performed using bootstrap resampling (Efron and Tibshirani, 1993). Except for the last row of the table, all tagging is performed without any text normalization. The last row demonstrates the upper bound performance on this task, by using both gold text normalization and gold part-of-speech tags. These results show that combining a generic part-of-speech tagger with a more coarse-grained domain-specific tagger can lead to measurable improvements in parse quality.

[3] `https://github.com/danieldk/jitar`
[4] `http://opennlp.apache.org/`

Tagging method	Unlabeled F_1	Labeled F_1
HMM	69.92	57.60
Maximum entropy	70.18	58.47
Coarse + n-best HMM	71.39*	58.39
Coarse + n-best MaxEnt	72.41*	60.16*
Gold norm., gold tags	79.28*	69.85*

* indicates statistical significance against MaxEnt baseline at p-value < 0.05.

Table 2: POS tagging and parse quality.

Normalization method	Unlabeled F_1	Labeled F_1
No normalization	72.41	60.16
+ Twitter syntax rules	76.17*	64.38*
Unsup. lexical	76.36*	64.80*
Machine translation	76.85*	65.38*
Unsup. lexical + MT	77.08*	65.57*
Gold norm., predic. tags[6]	78.20*	68.02*
Gold norm., gold tags	79.28*	69.85*

* statistically significant against non-normalized baseline at p-value < 0.05.

Table 3: Text normalization and parse quality.

5.2. Text Normalization

After considering the influence of the underlying POS tagger on parse quality, we now turn to the question of how much the parsing of noisy content is influenced by text normalization. For this, we evaluate two common text normalization methods: unsupervised normalization via lexical replacements and normalization based on machine translation.

Unsupervised lexical normalization Various unsupervised methods for text normalization have been suggested in the relevant literature. A popular approach is to perform lexical normalization by correcting individual tokens. We implement the model for lexical normalization of text messages by Han and Baldwin (2011). This method works in analogy to spell checking, with the biggest difference that in short message data ill-formedness is often intentional, for example due to the message size limit. The model performs normalization only on the token level.

Normalization as machine translation Research in short message normalization has shown that another effective method is to treat the task as a machine translation problem. Aw et al. (2006) and Raghunathan and Krawczyk (2009) explore phrase-based statistical machine translation as a preprocessing step for various NLP tasks involving text messages. As part of this effort, they manually normalize a set of 5.000 and 2.500 messages respectively. While these corpora are not created for social media services such as Twitter, they nonetheless provide reasonable training corpora for our experiments as the restrictions of both domains are similar.

Based on this corpus, we train a standard Moses baseline system[5] (Koehn et al., 2007) using GIZA++ for word alignments and the `grow-diag-final` symmetrization heuristic. An n-gram language model is built on the English side of the *news-commentary* data set using IRSTLM (Federico and Cettolo, 2007). Model weights are estimated using MERT (Och, 2003). All experiments are performed on the development part of our dataset.

Twitter-specific processing In order to isolate the influence of the text normalization, Twitter-specific syntax is parsed using a set of deterministic rules. Tokens such as retweet indicators and usernames at the start of a Tweet and URLs and hash tags at the end of a Tweet are removed from the text and pushed onto a stack. The remaining text is then parsed using the underlying dependency parser and the Twitter-specific tokens are re-attached to the tree according to a fixed set of rules. This deterministic handling of Twitter-specific syntax is applied to all further experiments in Table 3.

Impact on parse quality Table 3 presents the results of the text normalization schemes on the development part of our dataset. The results show that a combination of lexical and MT-based normalization approaches leads to results close to the upper bound set by gold standard normalization. Although the machine translation system was trained on a different domain, its application leads to better parsing results. This improved performance is most likely due to the fact that the method is able to normalize sequences of words on the phrase level instead of being restricted to single-word replacements.

6. Conclusion

User-generated content on the web constitutes a rich and important source of information for many use cases. However, parsing of such noisy data still poses challenges for many parsing algorithms. In this paper, we have compared various strategies for adapting dependency parsing to noisy input conditions. In order to do so, we introduced a noise-aware benchmark for dependency parsing consisting of a treebank and a corresponding evaluation metric. Our experiments on this new dataset show that text normalization improves parse quality significantly, especially if the normalization method can go beyond the word level (e.g. using machine translation). To encourage future progress in this area, we make available both the Denoised Web Treebank and the newly introduced noise-aware evaluation metric.[7]

Acknowledgments

We thank Gertjan van Noord for his valuable feedback. Parts of this work were supported through the Erasmus Mundus European Masters Program in Language and Communication Technologies (EM-LCT). The first author is supported by the EXPERT (EXPloiting Empirical appRoaches to Translation) Initial Training Network (ITN) of the European Union's Seventh Framework Programme. The second author is supported by the Nuance Foundation.

[5] `http://statmt.org/moses/?n=Moses.Baseline`

[6] Tags predicted by coarse + n-best MaxEnt.

[7] `http://jodaiber.de/DenoisedWebTreebank`

References

Aw, A., Zhang, M., Xiao, J., and Su, J. (2006). A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 33–40.

Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M. A., and Schasberger, B. (1995). Bracketing guidelines for Treebank II style Penn Treebank project. Technical report, University of Pennsylvania.

Bies, A., Mott, J., Warner, C., and Kulick, S. (2012). Bracketing webtext: An addendum to Penn Treebank II guidelines. Technical report, Linguistic Data Consortium.

Brants, T. (2000). TnT: a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA.

Buchholz, S. and Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Federico, M. and Cettolo, M. (2007). Efficient handling of n-gram language models for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 88–95.

Foster, J., Çetinoglu, Ö., Wagner, J., Le Roux, J., Nivre, J., Hogan, D., and van Genabith, J. (2011a). From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 893–901, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Foster, J., Çetinoglu, Ö., Wagner, J., Le Roux, J., Hogan, S., Nivre, J., Hogan, D., and Van Genabith, J. (2011b). #hardtoparse: POS Tagging and Parsing the Twitterverse. In *AAAI 2011 Workshop On Analyzing Microtext*, pages 20–25, United States.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (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.

Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 368–378.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech*. Pearson Education.

Kaljahi, R., Foster, J., Roturier, J., Ribeyre, C., Lynn, T., and Le Roux, J. (2015). Foreebank: Syntactic analysis of customer support forums. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1341–1347, Lisbon, Portugal, September.

Khan, M., Dickinson, M., and Kübler, S. (2013). Towards domain adaptation for parsing web data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 357–364, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, Bulgaria.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, N. A. (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.

Liu, X., Zhou, M., Wei, F., Fu, Z., and Zhou, X. (2012). Joint inference of named entity recognition and normalization for tweets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 526–535.

McDonald, R. T., Pereira, F., Ribarov, K., and Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing.*

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 160–167.

Petrov, S. and McDonald, R. (2012). Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, volume 59.

Prins, R. P. (2005). *Finite-state pre-processing for natural language analysis*. Ph.D. thesis, University of Groningen.

Raghunathan, K. and Krawczyk, S. (2009). Investigating SMS text normalization using statistical machine translation. *Technical report*.

Silveira, N., Dozat, T., de Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., and Manning, C. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland, May. European Language Resources Association (ELRA).