An In-depth Analysis of the Effect of Lexical Normalization on the Dependency Parsing of Social Media

Rob van der Goot
Center for Language and Cognition
University of Groningen
Department of Computer Science
IT University of Copenhagen
robv@itu.dk

Abstract

Existing natural language processing systems have often been designed with standard texts in mind. However, when these tools are used on the substantially different texts from social media, their performance drops dramatically. One solution is to translate social media data to standard language before processing, this is also called normalization. It is well-known that this improves performance for many natural language processing tasks on social media data. However, little is known about which types of normalization replacements have the most effect. Furthermore, it is unknown what the weaknesses of existing lexical normalization systems are in an extrinsic setting. In this paper, we analyze the effect of manual as well as automatic lexical normalization for dependency parsing. After our analysis, we conclude that for most categories, automatic normalization scores close to manually annotated normalization and that small annotation differences are important to take into consideration when exploiting normalization in a pipeline setup.

1 Introduction

It is well known that many traditional natural language processing systems are focused on standard texts, and their performance drops when used on another domain. This is also called the problem of domain adaptation. Recently, much focus has been on the notoriously noisy domain of social media. The hasty and informal nature of communication on social media results in highly non-standard texts, including a variety of phenomena not seen in standard texts, like phrasal abbreviations, slang, typos, lengthening, etc. One approach to adapt natural language processing tools to the social media domain is to 'translate' input to standard text before processing it, this is also referred to as normalization. In this approach, the input data is made more similar to the type of data the tool is expecting. Previous work has shown that normalization improves performance on social media data for tasks like POS tagging, parsing, lemmatization and named entity tagging (Baldwin and Li, 2015; Schulz et al., 2016; Zhang et al., 2013), however, it often remains unknown which types of replacements are most influential and which type of replacements still have potential to improve the usefulness of an automatic normalization system.

Baldwin and Li (2015) already investigated this effect in detail. They evaluate the effect of manual normalization beyond the word-level (including insertion and deletion of words). To the best of our knowledge, no automatic systems are available to obtain such a normalization, which is why Baldwin and Li (2015) focused only on the theoretical effect (i.e. manually annotated normalization). In this work, we will instead focus on lexical normalization, which is normalization on the word level. For this task, publicly available datasets and automatic systems are available (Han and Baldwin, 2011; Baldwin et al., 2015).

Recently, multiple English social media treebanks were released (Blodgett et al., 2018; Liu et al., 2018; van der Goot and van Noord, 2018) in Universal Depencies format (Nivre et al., 2017), as well as novel categorizations of phenomena occurring in lexical normalization (van der Goot et al., 2018). In this work, we combine both of these tasks into one dataset, which allows us not only to evaluate the theoretical effect of lexical normalization for dependency parsing, but also a real-world situation with automatic normalization.

The main contributions of this paper are:

- We add a layer of annotation to a social media treebank to also include normalization categories.

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 115–120

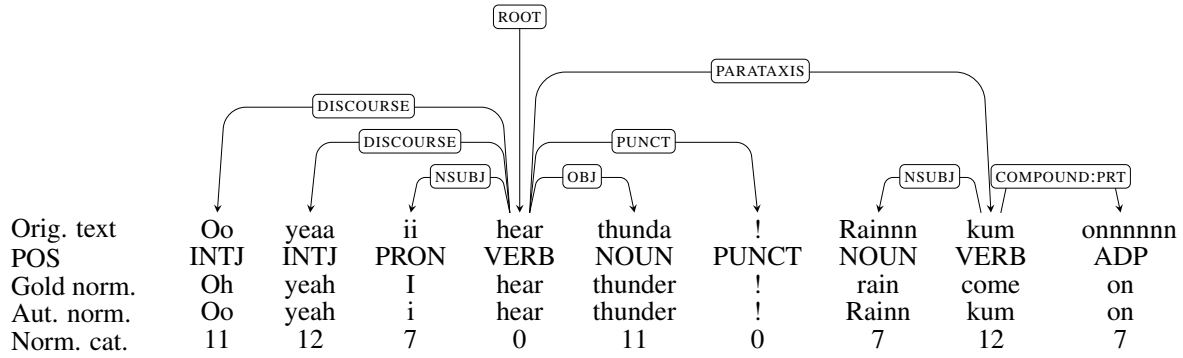

Figure 1: Example annotation for the sentence "Oo yeaa ii hear thunda ! Rainnn kum onnnnnn"

- We analyze the theoretical effect of lexical normalization for dependency parsing by using manually annotated normalization.
- We analyze the effect of an automatic lexical normalization model for dependency parsing, thereby showing which type of replacements still require attention.

2 Data

In this section we shortly discuss our choices for datasets and annotation formats, starting with the treebank data, followed by the lexical normalization categories annotation and automatic normalization. See Figure 1 for a fully annotated example instance from our development data.

2.1 Treebank

In 2018, three research groups simultaneously annotated dependency trees in the Universal Dependencies format on tweets: Liu et al. (2018) focussed on training a better parser by using an ensemble strategy, Blodgett et al. (2018) improved a dependency parser by using several adaptation methods, whereas van der Goot and van Noord (2018) focused on the use of normalization. Because the treebank created by van der Goot and van Noord (2018) is already annotated for lexical normalization, we will use this treebank.

The data from the treebank is taken from Li and Liu (2015), where van der Goot and van Noord (2018) only kept the tweets that were still available at the time of writing. The data from Li and Liu (2015) was in turn taken from two different sources: the LexNorm dataset (Han and Baldwin, 2011), originally annotated with lexical normalization and the dataset by Owoputi et al. (2013), originally annotated with POS tags. Li and Liu (2015) complemented this annotation so that both sets contain normalization as well as POS tags, to which van der Goot and van Noord (2018) added Universal Dependency structures. Similar to van der Goot and van Noord (2018) we use the English Web Treebank treebank (Silveira et al., 2014) for training, and Owoputi (development data) for the analysis. The test split is not used in this work, since our aim is not to improve the parser.

2.2 Normalization Categories

We choose to use the taxonomy of van der Goot et al. (2018) for three main reasons: 1) to the best of our knowledge, this is the most detailed categorization for lexical normalization 2) annotation for the same source data as the treebanks is available from Reijngoud (2019) 3) systems are available to automatically perform this type of normalization, as opposed to the taxonomy used by Baldwin and Li (2015). The existing annotation is edited to fit the treebank tokenization; if a word is split in the treebank, the normalization is split accordingly, and both resulting words are annotated in the same category. (Reijngoud, 2019) added one category to the taxonomy: informal contractions, which includes splitting of words like 'gonna' and 'wanna'. The frequencies of the categories in the development data are shown in Table 1. The 'split', 'merge' and 'phrasal abbreviations' categories are very infrequent, because the original annotation only included 1-1 replacements, these categories have been added when transforming the annotation to treebank tokenization.

2.3 Automatic Lexical Normalization

We use the state-of-the-art model for lexical normalization: MoNoise (van der Goot, 2019), which

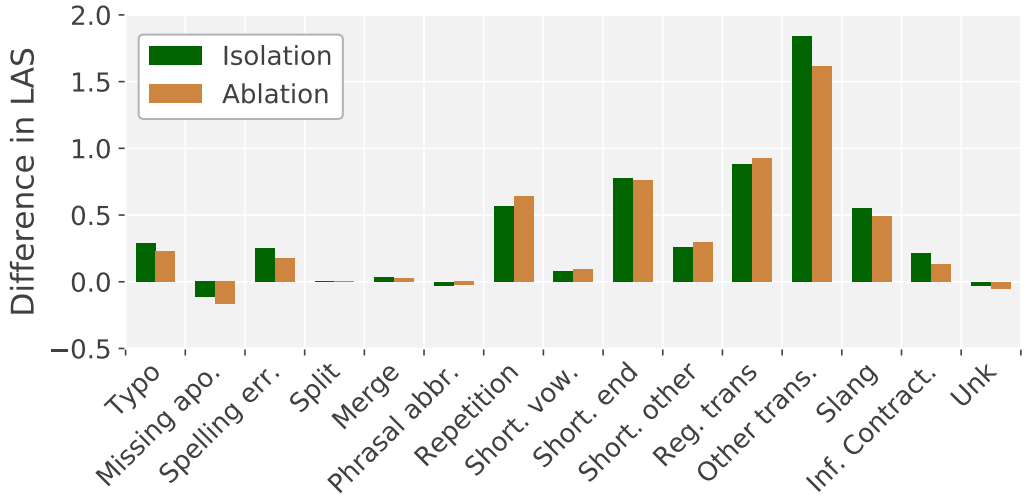

Figure 2: The effect of the categories when using manually annotated normalization. Isolation is the increase in performance when using only one category compared to using no normalization. Ablation is the loss when disabling only one category (higher is better).

Category	Freq.	%	Category	Freq.	%
No norm.	3,743	81.76	Short. vow.	22	0.48
Typo	30	0.66	Short. end	64	1.40
Missing apo.	176	3.84	Short. other	35	0.76
Spelling err.	44	0.96	Reg. trans.	66	1.44
Split	0	0.0	Other trans.	18	4.06
Merge	10	0.22	Slang	42	0.92
Phrasal abbr.	2	0.04	Inf. Contr.	56	1.22
Repetition	90	1.97	Unk	12	0.26

Table 1: Distribution of the replacement categories in the development data, 'No norm.' refers to words which are not normalized. For a detailed description of the categories we refer to (van der Goot et al., 2018).

is a modular normalization model, consisting of two steps; candidate generation and candidate ranking. For the generation, the most important modules are a lookup list based on the training data, the Aspell spell-checker[1] and word embeddings. For the ranking of candidates, features from the generation are complemented with n-gram probabilities and used as input to a random forest classifier, which predicts the confidence that a candidate is the correct replacement.

We train MoNoise on data from (Li and Liu, 2014), because it is most similar in annotation style to our development and test sets. Performance on the normalization task is slightly lower compared to the reported results (Error reduction rate (van der Goot, 2019) on the word level dropped from 60.61 to 45.38), because of differences in tokenization required for Universal Dependencies annotation. Also, the model clearly has issues with capitalization (see for example Figure 1) because capitalization is not corrected in the normalization training data.

3 Effect of Manual Normalization

We use the UUparser(de Lhoneux et al., 2017) for our experiments, with similar settings as van der Goot and van Noord (2018), including a heuristic to correctly parse a sentence starting with a retweet token or a username. All results reported in this paper are obtained with the official UD evaluation script[2] and are the average of 10 runs with different random seeds for the parser. For both settings (manual/automatic) we inspected the LAS graphs as well as the UAS graphs, but because the UAS scores showed very similar trends they are not reported here. The parser scores 52.56 LAS on the original input data, which improves to 57.83 when using the full gold normalization.

To evaluate the effect of each category, we measure performance twofold: in isolation, and in an ablation setting. For the isolation, we look at the difference between the baseline parser (without normalization) and a parser which only has access to normalization replacements of one category. For the ablation setting, we look at the loss when removing one category from the full model.

[1] http://aspell.net/

[2] http://universaldependencies.org/conll18/conll18_ud_eval.py

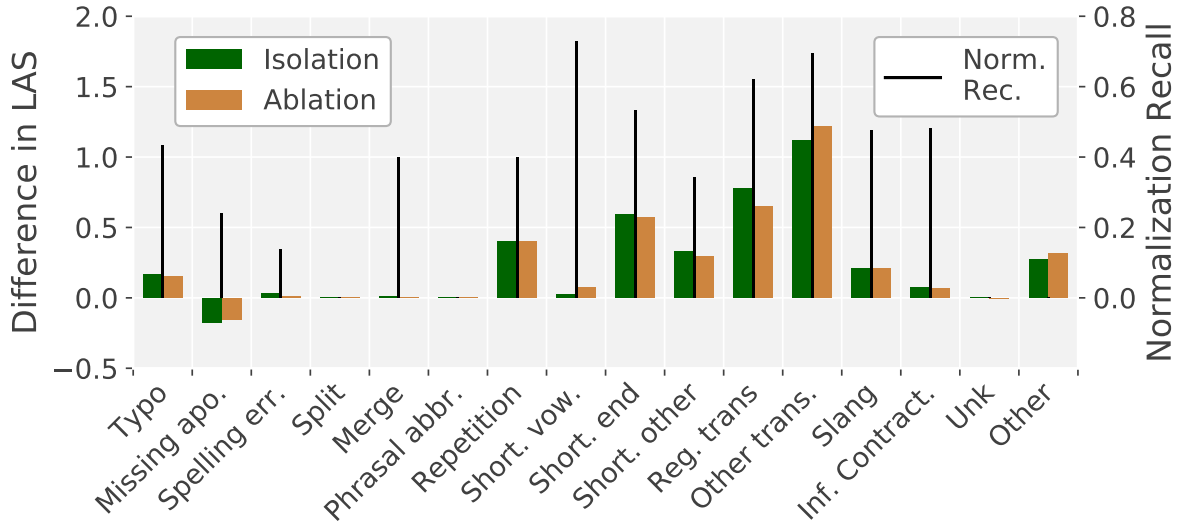

Figure 3: The effect of the categories when using automatic normalization. On the right y-axis the performance of the normalization model on this category is plotted (recall). The 'Other' category shows the effect of normalization replacements that were not annotated (but are still replaced by MoNoise).

The results for each category with gold normalization are shown in Figure 2. From these results, it becomes clear that some categories have a much larger effect compared to other categories. Not surprisingly, there is a correlation visible with the frequencies (Table 1). The categories going beyond the 1-1 normalization have only very little effect since they are very rare in this dataset[3]. The most important category is 'other transformation', this is mainly due to very frequent short words (e.g. 2↦to, u↦you). Other important categories are 'shortening end' and 'regular transformations'. This can be explained by the fact that they repair the suffixes, which often contain important syntactic clues.

It also becomes clear that differences in tokenization guidelines play a large role; one of the most frequent categories 'missing apostrophe' seems to be not useful for parsing; a manual inspection showed that this is because these also occur in the training data in their not-normalized form (e.g. 'll↦ will), thereby normalizing them creates more diversity. For the same reason, informal contractions (e.g. wanna, gonna) also have a relatively small effect.

4 Effect of Automatic Normalization

When using the full normalization model, the parser achieves a LAS of 56.32 when using all normalization categories, which is 72% of the gain that can be achieved with gold normalization compared to the baseline setting (52.56). Similar to the previous section, we run an isolation as well as an ablation experiment. In this setting, we only allow the normalization to replace words that are annotated as the category under evaluation (for the ablation experiments the inverse).

The parser performance as well as the recall of the normalization model on each category are plotted in Figure 3. Results show that the 'other transformations' and 'slang' category have the most room for improvement in LAS compared to gold normalization, even though they are not the worst categories with respect to the normalization performance. Furthermore, trends are rather similar compared to the gold normalization, even though there are differences in normalization performance. As expected from the gold normalization, the 'missing apostrophe' category is not helpful.

Interestingly, the 'other' category, which includes normalization replacements that were not annotated in the gold normalization, shows a small increase in performance. This category includes replacements like 'supp'↦'support' and 'da'↦'the', which were overlooked by the annotator. This could also be due to differences in the scope of annotation between the training data and development data.

[3]they were not annotated in their original releases, but were added when used in the treebank

5 Conclusion

We have introduced a novel annotation layer for an existing treebank with normalization annotation, which indicates which types of replacements are made. This allowed us to evaluate the effect of lexical normalization on the dependency parsing of tweets, both with manual normalization annotation and automatically predicted normalization. The automatic normalization obtained over 70% of the performance increase that could be obtained with gold normalization. The most influential categories were 'other transformation', which includes many replacements for very short words, and the categories with a high frequency that repair a words' suffix: 'shortening end' and 'regular transformation'. The categories which have the most potential for improvement in parser performance are the 'other transformation' and 'slang' categories. Furthermore, we saw that some predicted normalization replacements which were not annotated in the gold data also led to an increase in performance. Our results suggest that care should be taken when taking out-of-the-box annotation, because differences in annotation and the scope of the normalization task (i.e. tokenization, missed normalization) could lead to sub-optimal performance.

The dataset and code for the analysis is available on: https://bitbucket.org/robvanderg/taxeval/.

6 Acknowledgments

I would like to thank Wessel Reijngoud for providing the annotation of the normalization categories and Gertjan van Noord and the anonymous reviewers for their feedback.

References

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China. Association for Computational Linguistics.

Tyler Baldwin and Yunyao Li. 2015. An in-depth analysis of the effect of text normalization in social media. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 420–429, Denver, Colorado. Association for Computational Linguistics.

Su Lin Blodgett, Johnny Wei, and Brendan O'Connor. 2018. Twitter universal dependency parsing for African-American and mainstream American English. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Melbourne, Australia. Association for Computational Linguistics.

Rob van der Goot. 2019. MoNoise: A multi-lingual and easy-to-use lexical normalization tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, Florence, Italy. Association for Computational Linguistics.

Rob van der Goot and Gertjan van Noord. 2018. Modeling input uncertainty in neural network dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4984–4991, Brussels, Belgium. Association for Computational Linguistics.

Rob van der Goot, Rik van Noord, and Gertjan van Noord. 2018. A taxonomy for in-depth evaluation of normalization for user generated content. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the The 15th International Conference on Parsing Technologies (IWPT).*, Pisa, Italy.

Chen Li and Yang Liu. 2014. Improving text normalization via unsupervised model and discriminative reranking. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 86–93, Baltimore, Maryland, USA. Association for Computational Linguistics.

Chen Li and Yang Liu. 2015. Joint POS tagging and text normalization for informal text. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1263–1269.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into universal dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 965–975, New Orleans, Louisiana. Association for Computational Linguistics.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jörgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phuong Lê H`ông, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Luong Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Övrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia. Association for Computational Linguistics.

Wessel Reijngoud. 2019. Automatic classification of normalisation replacement categories: within corpus and cross corpora. Master's thesis, University of Groningen.

Sarah Schulz, Guy De Pauw, Orphée De Clercq, Bart Desmet, Véronique Hoste, Walter Daelemans, and Lieve Macken. 2016. Multimodular text normalization of Dutch user-generated content. *ACM Transactions on Intelligent Systems Technology*, 7(4):1–22.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Congle Zhang, Tyler Baldwin, Howard Ho, Benny Kimelfeld, and Yunyao Li. 2013. Adaptive parser-centric text normalization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1159–1168, Sofia, Bulgaria. Association for Computational Linguistics.