IHS_RD: Lexical Normalization for English Tweets

Dmitry Supranovich
IHS Inc. / IHS Global Belarus
131 Starovilenskaya St
220123, Minsk, Belarus
Dmitry.Supranovich@ihs.com

Viachaslau Patsepnia
IHS Inc.
55 Cambridge pkwy, Suite 601
Cambridge, MA 02142, USA
Slava.Patsepnia@ihs.com

Abstract

This paper describes the Twitter lexical normalization system submitted by IHS R&D Belarus team for the ACL 2015 workshop on noisy user-generated text. The proposed system consists of two components: a CRF-based approach to identify possible normalization candidates, and a post-processing step in an attempt to normalize words that do not have normalization variants in the lexicon. Evaluation on the test data set showed that our unconstrained system achieved the F-measure of 0.8272 (rank 1 out of 5 submissions for the unconstrained mode, rank 2 out of all 11 submissions).

1 Introduction

Social media texts found in such services as Twitter or Facebook have a great data-mining potential, as they offer real-time data that can be useful to monitor public opinion on brands, products, events, etc. However, current Natural Language Processing systems are usually optimized for clean data, which is not the type of data found in social media texts, as they are often noisy, containing a lot of slang, typos, and abbreviations.

Normalizing such text is challenging. We want to achieve high recall, making as many corrections as possible, but not at the expense of precision – words should not be incorrectly normalized.

Previous approaches to this task incorporated different tools and methods: dictionaries, language models, finite state transducers, and machine translation models. Some of the methods are unsupervised, though often requiring adjustment of parameters based on annotated data (Han and Baldwin (2011), Liu et al. (2011), and Gouws et al. (2011)). Some are supervised, like that in Chrupała (2014), making use of a Conditional Random Field (Lafferty et al., 2001) to learn the sequences of edit operations from labelled data.

In this paper, we present an approach based on the usage of normalization lexicons and a CRF model for identifying potential candidates.

2 Task Description

2.1 Dataset

The corpus provided by the organizers consists of 2950 annotated tweets. The annotations follow these guidelines (Baldwin et al., 2015):

- Non-standard words are normalized to one or more canonical English words based on a pre-defined lexicon. For instance, *l o v e* should be normalized to *love* (many-to-one normalization), *tmrw* to *tomorrow* (one-to-one normalization), and *cu* to *see you* (one-to-many normalization). Additionally, *IBM* should be left untouched as it is in the lexicon and it is in its canonical form, and the informal *lol* should be expanded to *laughing out loud*.
- Non-standard words may be either out-of-vocabulary (OOV) tokens (e.g., *tmrw* for *tomorrow*) or in-vocabulary (IV) tokens (e.g., *wit* for *with* in "I will come wit you").
- Only alphanumeric tokens (e.g., *2*, *4eva* and *tmrw*) and apostrophes used in contractions (e.g., *yoou've*) are considered for normalization. Tokens including hyphens, single quotes and other types of contractions should be ignored.
- Domain specific entities are ignored even if they are in non-standard forms, e.g., *#ttyl*, *@nyc*
- It is possible for a tweet to have no non-standard tokens but still require normalization (e.g., the example of *wit* above), and it is also possible for the tweet to require no normalization whatsoever.

Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, pages 78–81,
Beijing, China, July 31, 2015.

- Proper nouns should be left untouched, even if they are not in the given lexicon (e.g., *Twitter*).
- All normalizations should use the American spelling (e.g., *tokenize* rather than *tokenise*).

2.2 Evaluation

Evaluation was to be carried out according to Precision, Recall, and F1 metrics.

3 Experimental Setup

First, a normalization lexicon was generated from the given training data, enriched with the data from several sources:

- Word pairs extracted from the datasets used for lexical normalization (Han, 2011; Liu, 2011)
- The online social media abbreviation list of Beal (2015)[1]. Compared to the previous workshops with one-to-one normalizations, the current task also considers one-to-many normalizations, and obviously not all abbreviations are present in the training data, so the use of a list of social media abbreviations can be vital to the system.

At the current stage of development the system is unable to differentiate between several normalization variants; thus, entries with multiple possible variants were reviewed to make the most suitable variant first in the list (entries that are most frequent in datasets are placed first, any ties were manually reviewed).

Second, a CRF model was trained. The labels chosen were CAND and NOT_CAND, reflecting potential normalization candidates and words that should not be normalized, respectively. The following features were used:

Token: This feature represents the string of the current token.

Context Feature: The token to the left and the token to the right are used as two context features. The surrounding words usually convey useful information about a token which helps in predicting the correct tag for each token.

Alphanumeric feature: This feature checks whether the token adheres to the annotation guidelines and makes sure that non-adhering tokens are not marked as potential candidates.

Normalization dictionary feature: This feature checks whether the token is present in the generated normalization lexicon.

Canonical lexicon feature: This feature indicates whether or not the token is present in the canonical lexicon provided by the workshop organizers.

Word length and number of vowels: Two separate features as well as their correlation, allowing to tag words with uncommon length-vowel correlation, like *bcz, pls,* etc.

Edit distance feature: marks a token that is within an edit distance of 2 or less from any word in the canonical lexicon.

Third, the text is normalized:

- All tokens tagged as potential candidates by the CRF model are normalized to their lexicon variants.
- All alphanumeric words are normalized to the American spelling with the VarCon tool (Atkinson, 2015)[2]. This includes the tokens which are already normalized using the lexicon.
- We have also tried to improve the normalization results by using a did-you-mean (DYM) module that is currently being developed at IHS R&D team. The DYM module corrects user queries/sentences with misspellings by providing corrected variant(s) with a confidence measure (including no correction variant with the corresponding confidence measure). The DYM module is an SVM model trained on a set of features for each of the multiple candidates generated for an input query/sentence. We used the following features: error model score, Levenshtein distance, language model score, the ratio of common noun vocabulary words, the ratio of proper noun vocabulary words, and the number of changes in non-lowercase words. An error model score was obtained from an autocompletion and autocorrection module (AAM) for which an index was built from 12.4M documents (scientific papers - 42.1%, Wikipedia articles - 23.5%, patents - 19.4%, social texts - 8%, and news - 7%). The 2-gram language model was built from 177K patents (1.36G words and 2.6M vocabulary). Since we did not have enough time to tailor both DYM and AAM modules for social text processing, DYM and AAM modules were

[1] http://www.webopedia.com/quick_ref/textmessageabbreviations.asp

[2] http://wordlist.aspell.net/varcon/

used for this Twitter lexical normalization system as is, being actually tailored for technical and scientific texts.

3.1 Results and error analysis

Testing was performed on the provided corpus of 1967 tweets.

Table 1 shows the performance of our CRF candidate model with different features:

- A baseline model with only token, context and alphanumeric features.
- A baseline model with the normalization dictionary and the canonical lexicon features added.
- A model with all features enabled.

Table 2 reflects our submitted normalization result and a result without the DYM module described above.

	Precision (CRF | Final)	*Recall (CRF | Final)*	*F1 (CRF | Final)*
Tokens + Context + Alphanumeric	0.991 | 0.8782	0.57 | 0.6013	0.7237 | 0.7139
Added dictionary features	0.907 | 0.8376	0.824 | 0.8133	0.8635 | 0.8253
All features	0.915 | 0.8469	0.817 | 0.8083	0.8632 | **0.8272**

Table 1. Result metrics of candidate CRF model with different features (and its impact on the result after normalization using a submitted system).

	Precision	*Recall*	*F1*
Lexicon Normalization + DYM (submitted)	0.8469	0.8083	0.8272
Lexicon Normalization without DYM	0.8765	0.7949	**0.8337**

Table 2. Result metrics of two normalization system configurations.

The DYM feature does a good job correcting typos and removing excessive duplicate letters (*beutiful* → *beautiful*, *tosee* → *to see*, and *smileeeeee* → *smile*). However, even with a high confidence threshold, quite a number of words are normalized excessively, mainly those in non-English (or partially English) tweets, e.g. *jeil* → *jail*, *hoje* → *hope*, and *wasan* → *was an*, in addition to some incorrect normalizations like *parkd* → *park* (instead of *parked*) or *hundread* → *hundreds* (instead of *hundred*). These mistakes are frequent, and an increase in recall does not outweigh a loss in precision; thus, the F-measure without the DYM feature in its current state is even a little bit higher than our submitted system with it. Lowering the confidence threshold brings more correct normalizations, but due to the nature of tweets even more incorrect ones, leading to an overall drop in F1 score. Nevertheless, we decided to use and submit the system with DYM, since we believe the text normalized this way is more suitable for further use.

Attempts were made to improve the performance of the DYM module as well as to select the correct candidate from a normalization lexicon if there is more than one variant present (ur → you're, your, you). For example, language detection works well on regular search queries and could potentially forbid the normalization of words in non-English tweets. However, it proved to be not helpful for tweets – the messages are short, some of them are a mixture of English and some other language (thus, if there is a normalization restriction on such tweets, potential English normalizations are lost), and slang- and abbreviation-rich tweets are hard to analyse. A language model was used in an attempt to select a correct normalization from multiple variants, but this did not prove to be effective, likely because the model used was not focused on social media texts.

We see room for potential improvement in tuning the DYM tool to social media texts, as well as in filtering non-English words from normalization candidates, experimenting with language models tailored to social media texts and further enriching the lexicon with new normalization data.

4 Conclusion

In this paper, we presented a system designed for participation in shared task #2 of the ACL 2015 workshop on noisy user-generated text. Our system makes use of CRF for identifying potential candidates, lexicons to normalize them and a DYM module as a post-processing step to further correct some of the misspelled words. Our system ranked second among all 11 submissions with 0.8272 F-measure and ranked first among 5 submissions for the unconstrained mode.

References

Kevin Atkinson. VarCon. Vers. 2015.02.15. *Web*. 01 Apr. 2015. http://wordlist.aspell.net/varcon/

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 (WNUT 2015)*, Beijing, China.

Vangie Beal. Text messaging and online chat abbreviations. *Web*. 01 Apr. 2015. http://www.webopedia.com/quick_ref/textmessageabbreviations.asp

Grzegorz Chrupała. 2014. Normalizing tweets with edit scripts and recurrent neural embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 680–686, Baltimore, USA.

Stephan Gouws, Dirk Hovy, and Donald Metzler. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90, Edinburgh, UK.

Bo Han and Timothy Baldwin. 2011. Lexical normalization of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 368-378, Portland, USA.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML'01, pages 282–289. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011. Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 71-76, Portland, USA.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 1035-1044, Jeju Island, Korea.

Yi Yang and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *Proceedings of Conference on Empirical Methods in Natura Language Processing (EMNLP)*, pages 61-72, Seattle, USA.