USZEGED: Correction Type-sensitive Normalization of English Tweets Using Efficiently Indexed n-gram Statistics

Gábor Berend
University of Szeged
Department of Informatics
Árpád tér 2., 6720 Szeged, Hungary
berendg@inf.u-szeged.hu

Ervin Tasnádi
University of Szeged
Department of Informatics
Árpád tér 2., 6720 Szeged, Hungary
Tasnadi.Ervin@stud.u-szeged.hu

Abstract

This paper describes the framework applied by team USZEGED at the "Lexical Normalisation for English Tweets" shared task. Our approach first employs a CRF-based sequence labeling framework to decide the kind of corrections the individual tokens require, then performs the necessary modifications relying on external lexicons and a massive collection of efficiently indexed n-gram statistics from English tweets. Our solution is based on the assumption that from the context of the OOV words, it is possible to reconstruct its IV equivalent, as there are users who use the standard English form of the OOV word within the same context. Our approach achieved an F-score of 0.8052, being the second best one among the unconstrained submissions, the category our submission also belongs to.

1 Introduction

Social media is a rich source of information which has been proven to be useful to a variety of applications, such as event extraction (Sakaki et al., 2010; Ritter et al., 2012; Ritter et al., 2015) or trend detection, including the tracking of epidemics (Lamb et al., 2013). Analyzing tweets in general, however, can pose several difficulties. From an engineering point of view, the streaming nature of tweets requires that special attention is paid to the scalability of the algorithms applied and from an NLP point of view, the often substandard characteristics of social media utterances has to be addressed. The fact that tweets are often written on mobile devices and are informal makes the misspelling and abbreviations of words and expressions, as well as the use of creative informal language prevalent, giving rise to a higher number of out-of-vocabulary (OOV) words than in other genres.

2 Related Work

The informal language of social media, including Twitter, is extremely heterogeneous, making its grammatical analysis more difficult compared to standard genres such as newswire. It has been shown previously, that the performance of linguistic analyzers trained on standard text types degrade severely once they are applied to texts found in social media, especially tweets (Ritter et al., 2011; Derczynski et al., 2013).

In order to build taggers that perform more reliable on social media texts, one possible way is to augment the training data by including texts originating from social media (Derczynski et al., 2013). Such approaches, however, require considerable human effort, so one possible alternative can be to normalize the social media texts first, then apply standard analyzers on these normalized texts. Recently, a number of approaches have been proposed for the lexical normalization of informal (mostly social media and SMS) texts (Liu et al., 2011; Liu et al., 2012; Han et al., 2013; Yang and Eisenstein, 2013).

Han and Baldwin (2011) rely on the identification of the words that require correction, then define a confusion set containing the candidate IV correction forms for such words. Finally, a ranking scheme, taking multiple factors into consideration, is applied which selects the most likely correction for an OOV word. In their subsequent work, Han et al. (2012) propose an automated method to construct accurate normalization dictionaries.

Liu et al. (2011; 2012) propose a character-level sequence model to predict insertions, deletions and substitutions. They first collect a large set of noisy `(OOV, IV)` training pairs from the Web. These pairs are then aligned at the character level

Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, pages 120–125,
Beijing, China, July 31, 2015.

and provided as training data for a CRF classifier. The authors also released their 3,802-element normalization dictionary that our work also relies at.

Yang and Eisenstein (2013) introduce an unsupervised log-linear model for the task of text normalization. Besides the features that can be derived from pairs of words (e.g. edit distance), features considering the context are also employed in their model. As the number of class labels in that model is equal to the size of the IV words an OOV word could possibly be corrected to (typically on the order of 10^4-10^5, which is far beyond the typical label size of classification tasks), the authors propose the use of Sequential Monte Carlo training approach for learning the appropriate feature weights.

3 The Task of Lexical Normalization

Formally, given an m-long sequence of words in the i^{th} tweet, $T_i = [t_{i,1}, t_{i,2}, \ldots, t_{i,m}]$, participants of the shared task had to return a sequence of normalized in-vocabulary (IV) words, i.e. $S_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,m}]$. The training set of the shared task consisted of 2,950 tweets comprising 44,385 tokens, while the test set had 1,967 tweets which included a total of 29,421 tokens. According to the dataset, most of the words did not require any kind of corrections, i.e. the proportion of unmodified words was 91.12% and 90.57% for the training and test set, respectively. Further details with respect the shared task can be found in the paper (Baldwin et al., 2015).

As a consequence, we first built a sequence model to decide *which* tokens need to be corrected and *in what way*. A typical distinction of the correction types would be based on the number of tokens a noisy token and its corrected form comprises of. According to this approach, one could distinguish between one-to-one, one-to-many and many-to-one corrections on the per token basis. However, instead of applying the above types of corrections, we identified a more detailed categorization of the correction types and trained a linear chain CRF utilizing CRFsuite (Okazaki, 2007). The correction types a token could be classified as were the following:

- $MissingApos$, standing for tokens that only differ from their corrected version in the absence of an apostrophe (e.g. *youll* → *you'll*),
- $MissingWS$, standing for tokens that only differ from their corrected version in the absence of one or more whitespace characters (e.g. *whataburger* → *what a burger*),
- $1to1_{ED\leq 2}$, standing for corrections where no whitespace characters had to be inserted and the augmented edit distance (introduced in Section 4.2) between the noisy token and its normalized form was at most 2 (e.g. *tmrw*→*tomorrow*),
- $1to1_{ED\geq 3}$, standing for corrections where no whitespace characters had to be inserted and the augmented edit distance was at least 3 (e.g. *plz*→*please*),
- $1toM_{ABB}$, standing for corrections where both whitespace and alphanumeric characters had to be inserted to obtain a tokens corrected variant (e.g. *lol* → *laugh out loud*).

	Training	Test
$MissingApos$	507	369
$MissingWS$	126	76
$1to1_{ED\leq 2}$	1,979	1,405
$1to1_{ED\geq 3}$	413	292
$1toM_{ABB}$	917	634
$Subtotal$	3,942	2,776
O	40,443	26,645
$Total$	44,385	29,421

Table 1: Distribution of the correction types in the training and test sets

For the sake of completeness, we should add that a further class label (*O*) was employed. This, however, corresponded to the case when there was no correction required to be performed for a token. As mentioned above, more than 90% of the words in both the training and test sets belonged to this category. Table 1 shows the distribution of the correction types on both the training and test sets.

4 Proposed Approach

Our approach consists of a sequence labeling module and relies on lookups from an efficiently indexed n-gram corpus of English tweets. Subsequently, we describe the details of these modules.

4.1 Sequence Labeling for Determining Correction Types

As already mentioned in Section 3, the first component in our pipeline was a linear chain CRF

(Lafferty et al., 2001). Besides the common word surface forms, such as the capitalization pattern, the first letter or character suffixes, we relied on the following dictionary resources upon determining the features for the individual words:

- the SCOWL dictionary being part of the `aspell` spell checker project containing canonical English dictionary entries,
- the normalization dictionaries of Han et al. (2012) and Liu et al. (2012),
- the 5,307-element normalization dictionary derived from the portal `noslang.com`, which map common social media abbreviations to their complete forms.

For each token, word type features were generated along with the word types of its neighboring tokens. The POS tags assigned to each token and its neighboring tokens by the Twitter POS tagger (Gimpel et al., 2011) were also utilized as features in the CRF model. The Twitter POS tag set was useful to us, as it contains a separate tag (**G**) for multi-word abbreviations (e.g. *ily* for *I love you*), which was expected to be highly indicative for the correction type $1toM_{ABB}$.

In order to be able to discriminate the $MissingWS$ class, we introduced a feature which indicates for a token t originating from a tweet whether the relation

$$\max_{s \in split(t)} freq_{1T}(s) \geq \tau$$

holds, where τ is a threshold calibrated to 10^6 based on the training set, $freq_{1T}(s)$ is a function which returns the frequency value associated with a string s according to the Google 1T 5-gram corpus and the function $split(t)$ returns the set of all the possible splits of token t such that its components are all contained in the SCOWL dictionary. For instance $split(\text{“}whataburger\text{”})$ returns a set of splits including “*what a burger*”, “*what a burg er*” and “*what ab urger*”. As there is a split (i.e. “*what a burger*”) that is sufficiently frequent according to the n-gram corpus, we take it as an indication that the original token omitted some whitespace characters that we need to inserted.

A CRF model with the above feature set was trained using L-BFGS training method and L1 regularization using CRFsuite (Okazaki, 2007). The overall token accuracy this model achieved was 0.9830 and 0.9746 and the proportion of tweets for which all the tokens were tagged properly was 0.7902 and 0.7143 for the training and test sets, respectively. A more detailed breakdown of the classification performances of the sequence model on the training and test sets are included in Table 2 and Table 3. These tables reveal that the most difficult error type to identify was the one where a word missed some whitespace characters (row $MissingWS$). This class happens to be the least frequent and one of the most heterogeneous class as well, which might be an explanation for the lower results on that class.

	Precision	Recall	F-score
$MissingApos$	0.9686	0.9744	0.9715
$MissingWS$	0.8795	0.5794	0.6986
$1to1_{ED \leq 2}$	0.9078	0.8504	0.8782
$1to1_{ED \geq 3}$	0.9593	0.6852	0.7994
$1toM_{ABB}$	0.9624	0.8942	0.9271
O	0.9874	0.9959	0.9916
macro average	0.9442	0.8299	0.8777

Table 2: Results of predicting the correction types for tokens on the training set

	Precision	Recall	F-score
$MissingApos$	0.9755	0.9702	0.9728
$MissingWS$	0.7674	0.4342	0.5546
$1to1_{ED \leq 2}$	0.8619	0.7950	0.8271
$1to1_{ED \geq 3}$	0.8793	0.5240	0.6567
$1toM_{ABB}$	0.9449	0.8659	0.9037
O	0.9816	0.9932	0.9874
macro average	0.9018	0.7638	0.8171

Table 3: Results of predicting the correction types for tokens on the test set

4.2 Augmented Edit Distance

When determining a set of candidate IV words that an OOV might be rewritten for, it is a common practice to place an upper bound on the edit distance between the IV candidates and the OOV word. In order to measure edit distance between tokens originating from tweets and their corrected forms, we implemented a modification of the standard edit distance algorithm that is especially tailored to measuring the difference of OOV tokens originating from social media to IV ones.

The edit distance we employed is asymmetric as insertions of characters into OOV tokens have no costs. For instance, for the words *tmrw* and *to-*

morrow, the edit distance is regarded as 0 if the former is considered to be the substandard OOV token and the latter one as the standard IV one. Note, however, if the role of the two tokens was changed (i.e if *tmrw* was treated as IV and *tomorrow* as OOV), their edit distance would become 4. A further relaxation to the standard edit distance is that we assign 0 cost to the following kinds of phonetically motivated transcriptions:

- $z \rightarrow s$ located at the end of words (e.g. in $cat\underline{z} \rightarrow cat\underline{s}$),
- $a \rightarrow er$ located at the end of words (e.g. in $bigg\underline{a} \rightarrow bigg\underline{er}$).

By making the above relaxations to the definition of the standard edit distance, we could obtain larger candidate sets for a given edit distance threshold for tokens with higher recall, as we could reduce the edit distance between the OOV words and their appropriate IV equivalent in many cases. Obviously, as the candidate set grows, it might get increasingly difficult to choose the correct normalization from it. However, at this stage of our pipeline, we were more interested in having the correct IV word in the set of candidate normalization, rather than reducing its size.

4.3 Making Use of Twitter n-gram Statistics

Our basic assumption was that from the context of an OOV word, it is possible to reconstruct its IV equivalent, as there are users who use the correct IV English form of the OOV word within the same context, e.g. *see you tomorrow* instead of *see u tmrw*. The Twitter n-gram frequencies we made use of were the ones that we aggregated over the Twitter n-gram corpus augmented with demographic metadata described in (Herdadelen, 2013).

For a given token t_i at position i in a tweet, we chose the most probable corrected form according to the formula

$$\underset{t' \in C(t_i, ct(t_i))}{\arg\max} P(t'|t_{i-1})P(t_{i+1}|t'), \qquad (1)$$

where the function $C(t_i, ct(t_i))$ returns a set of IV candidates for the token t_i, according to $ct(t_i)$, which is the correction type determined for that token by the sequence model introduced in Section 4.1. We indexed the Twitter n-gram corpus with the highly effective LIT indexer (Ceylan and Mihalcea, 2011), which made fast queries of the form $\mathtt{t_{i-1}} * \mathtt{t_{i+1}}$ possible, the symbol $*$ being a placeholder for any token at the given position. The only case when we did not choose the normalization of an OOV word according to (1) was when there was a unique suggestion for an IV word in the normalization dictionaries we listed in Section 4.1.

Correction	Precision	Recall	F-score
$MissingApos$	0.9972	0.9972	0.9972
$MissingWS$	0.8684	0.4177	0.5641
$1to1$	0.9191	0.9219	0.9205
$1toM_{ABB}$	0.8861	0.9533	0.9185

Table 4: Detailed performance on the different correction types on the training dataset

Correction	Precision	Recall	F-score
$MissingApos$	1.0000	0.9841	0.992
$MissingWS$	0.9737	0.4458	0.6116
$1to1$	0.9141	0.9127	0.9134
$1toM_{ABB}$	0.8523	0.9699	0.9073

Table 5: Detailed performance on the different correction types on the test dataset

The performance of the normalization on the training and test sets, according to the correction types we defined can be found in Table 4 and Table 5, respectively. From these tables, one can see that the worst results were obtained for the correction type when spaces were required to be inserted to a OOV word.

This is in accordance with the fact that our sequence model obtained the lowest scores exactly on this kind of corrections. However, due to the fact that this error category is the least frequent, the lower scores on that category does not harm that much our overall performance as can be seen in Table 6 for both the training and test corpora. The results shown in Table 6 also illustrate that our approach seems to generalize well, as there is a small gap between the performances observed on the training and test sets of the shared task.

	Training	Test
precision	0.8703	0.8606
recall	0.7673	0.7564
F1	**0.8156**	**0.8052**

Table 6: Overall performance of our system on the training and test sets

5 Conclusion

In this paper, we introduced our approach to the lexical normalization of English tweets that ranked second at the shared task among the unconstrained submissions. Our framework first performs sequence labeling over the tokens of a tweet to predict *which* tokens need to be corrected and in *what way*. This step is followed by correction type-sensitive candidate set generation, from which set the most likely IV normalization of an OOV word is selected by querying an efficiently indexed large n-gram dataset of English tweets.

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 (WNUT 2015)*, Beijing, China.

Hakan Ceylan and Rada Mihalcea. 2011. An efficient indexer for large n-gram corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, HLT '11, pages 103–108, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. 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.

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 - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 421–432, Jeju Island, Korea.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Trans. Intell. Syst. Technol.*, 4(1):5:1–5:27, February.

Ama Herdadelen. 2013. Twitter n-gram corpus with demographic metadata. *Language Resources and Evaluation*, 47(4):1127–1147.

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, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Alex Lamb, Michael J. Paul, and Mark Dredze. 2013. Separating fact from fear: Tracking flu infections on twitter. In *In NAACL*.

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: Human Language Technologies*, pages 71–76, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A Broad-Coverage Normalization System for Social Media Language. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1035–1044.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1104–1112, New York, NY, USA. ACM.

Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. 2015. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 896–905, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

Yi Yang and Jacob Eisenstein. 2013. A Log-Linear Model for Unsupervised Text Normalization. *Proceedings of the Empirical Methods on Natural Language Processing (EMNLP)*, pages 61–72.