Standardizing Tweets with Character-level Machine Translation

Nikola Ljubešić[1], Tomaž Erjavec[2], and Darja Fišer[3]

[1]University of Zagreb, Faculty of Humanities and Social Sciences, Zagreb, Croatia
nikola.ljubesic@ffzg.hr
[2] Jožef Stefan Institute, Department of Knowledge Technologies, Ljubljana, Slovenia
tomaz.erjavec@ijs.si
[3]University of Ljubljana, Faculty of Arts, Ljubljana, Slovenia
darja.fiser@ff.uni-lj.si

Abstract. This paper presents the results of the standardization procedure of Slovene tweets that are full of colloquial, dialectal and foreign-language elements. With the aim of minimizing the human input required we produced a manually normalized lexicon of the most salient out-of-vocabulary (OOV) tokens and used it to train a character-level statistical machine translation system (CSMT). Best results were obtained by combining the manually constructed lexicon and CSMT as fallback with an overall improvement of 9.9% increase on all tokens and 31.3% on OOV tokens. Manual preparation of data in a lexicon manner has proven to be more efficient than normalizing running text for the task at hand. Finally we performed an extrinsic evaluation where we automatically lemmatized the test corpus taking as input either original or automatically standardized wordforms, and achieved 75.1% per-token accuracy with the former and 83.6% with the latter, thus demonstrating that standardization has significant benefits for upstream processing.

Keywords: twitterese, standardization, character-level machine translation

1 Introduction

This paper deals with the problem of processing non-standard language for smaller languages that cannot afford to develop new text processing tools for each language variety. Instead, language varieties need to be standardized so that the existing tools can be utilized with as little negative impact of the noisy data as possible. Slovene, the processing of which is difficult already due to its highly inflecting nature, is even harder to process when orthographic, grammatical and punctuation norms are not followed. This is often the case in non-standard and less formal language use, such as in the language of tweets which is becoming a predominant medium for the dissemination of information, opinions and trends and as such an increasingly important knowledge source for data mining and text processing tasks. Another important characteristics of twitterese is that it is rich

in colloquial, dialectal and foreign-language elements, causing the standard text processing tools to underperform.

This is why we propose an approach to standardizing Slovene tweets with the aim of increasing the performance of the existing text processing tools by training a character-level statistical machine translation (CSMT) system. CSMT has recently become a popular method for translating between closely related languages, modernizing historical lexicons, producing cognate candidates etc. The specificity of CSMT is that the translation and language model are not built from sequences of words, but characters. In all experiments we use the well-known Moses system[1] with default settings if not specified differently. In order to minimize the human input required, we explore the following strategy: we produce a manually validated lexicon of the 1000 most salient out-of-vocabulary (OOV) tokens in respect to a reference corpus, where the lexicon contains pairs (original wordform, standardized wordform). We also annotate a small corpus of tweets with the standardized wordform and use the lexicon resource for training the CSMT system and the corpus for evaluating different settings. We compare the efficiency of normalizing a lexicon of most-salient OOV tokens to the standard approach of normalizing running text. Finally, we also manually lemmatize our test corpus in order to evaluate how much the standardization helps with the task of lemmatization. The datasets used in this work are made available together with the paper[2].

The rest of this paper is structured as follows: Section 2 discusses related work, Section 3 introduces the dataset we used for the experiments, Section 4 gives the experiments and results, while Section 5 concludes and gives some directions for future work.

2 Related work

Text standardization is rapidly gaining in popularity because of the explosion of user-generated text content in which language norms are not followed. SMS messages used to be the main object of text standardization [2, 3] while recently Twitter has started taking over as the most prominent source of information encoded with non-standard language [7, 6].

There are two main approaches to text standardization. The unsupervised approach mostly relies on phonetic transcription of non-standard words to produce standard candidates and language modeling on in-vocabulary (IV) data for selecting the most probable candidate [6]. The supervised approach assumes manually standardized data from which standardization models are built.

Apart from using standard machine learning approaches to supervised standardization, such as HMMs over words [3] or CRFs for identifying deletions [8], many state-of-the-art supervised approaches rely on statistical machine translation which defines the standardization task as a translation problem. There has been a series of papers using phrase-based SMT for text standardization [2,

[1] http://www.statmt.org/moses/
[2] http://www.cicling.org/2014/data/156/

7] and, to the best of our knowledge, just two attempts at using character-level SMT (CSMT) for the task [9, 4]. Our work also uses CSMT but with a few important distinctions, the main one being data annotation procedure. While [9, 4] annotate running tweets, we investigate the possibility of extracting a lexicon of out-of-vocabulary (OOV) but highly salient words with respect to a reference corpus. Furthermore, we apply IV filters on the n-best CSMT hypotheses which proved to be very efficient in the CSMT approach to modernizing historical texts [11]. Finally, we combine the deterministic lexicon approach with the CSMT approach as fallback for tokens not covered by the lexicon.

3 Dataset

The basis for our dataset was the database of tweets from the now no longer active aggregator sitweet.com containing (mostly) Slovene tweets posted between 2007-01-12 and 2011-02-20. The database contains many tweets in other languages as well, so we first used a simple filter that keeps only those that contain one of the Slovene letters č, š or ž. This does not mean that there is no foreign language text remaining, as some closely related languages, in particular Croatian, also use these letters. Also it is fairly common to mix Slovene and another language, mostly English, in a single tweet. However, standard methods for language identification do not work well with the type of language found in tweets, and are also bad at distinguishing closely related languages, especially if a single text uses more than one language. In this step we also shuffled the tweets in the collection so that taking any slice will give a random selection of tweets, making it easier to construct training and testing datasets.

In the second step we anonymized the tweets by substituting hashtags, mentions and URLs with special symbols (XXX-HST, XXX-MNT, XXX-URL) and substituted emoticons with XXX-EMO. This filter is meant to serve two purposes. On the one hand, we make the experimental dataset freely available and by using rather old and anonymized tweets we hope to evade problems with the Twitter terms of use. On the other, tweets are difficult to tokenize correctly and by substituting symbols for the most problematic tokens, i.e. emoticons, we made the collection easier to process.

We then tokenized the collection and stored it in the so called vertical format, where each line is either an XML tag (in particular, `<text>` for an individual tweet) or one token. With this we obtained a corpus of about half a million tweets and eight million word tokens which is the basis for our datasets.

3.1 Support lexicons

As will be discussed in the following sections, we also used several support lexicons to arrive at the final datasets for our experiments. In the first instance, this is Sloleks[3] [1], a CC-BY-NC available large lexicon of Slovene containing

[3] http://eng.slovenscina.eu/sloleks/opis

the complete inflectional paradigms of 100,000 Slovene lemmas together with their morphosyntactic descriptions and frequency of occurrence in the Gigafida reference corpus of Slovene. We used only wordforms and their frequency from this lexicon, not making use of the other data it contains. In other words, to apply the method presented here to another language only a corpus of standard language is needed, from which a frequency lexicon, equivalent to the one used here, can then be extracted.

As mentioned, Slovene tweets often mix other languages with Slovene and, furthermore, the language identification procedure we used is not exact. As processing non-Slovene words was not the focus of this experiment, it was therefore useful to be able to identify foreign words. To this end, we made a lexicon of words in the most common languages appearing in our collection, in particular English and Croatian. For English we used the SIL English wordlist[4], and for Croatian the lexicon available with the Apertium MT system[5].

A single lexicon containing all three languages was produced, where each wordform is marked with one or more languages. It is then simple to match tweet wordforms against this lexicon and assign each such a word a flag giving the language(s) it belongs or marking it as OOV.

3.2 Lexicon of Twitterese

The most straightforward way to obtain standardizations of Twitter-specific wordforms is via a lexicon giving the wordform and its manually specified standardized form. If we choose the most Twitter-specific wordforms, this will cover many tokens in tweets and also take care of some of the more unpredictable forms.

To construct such a lexicon, we first extracted the frequency lexicon from the tweet corpus vertical file. We then used Sloleks to determine the 1,000 most tweet-specific words using the method of frequency profiling [10] which, for each word, compares its frequency in the specialized corpus to that in the reference corpus using log-likelihood. These words were then manually standardized, a process that took about three hours, i.e. on the average about 10s per entry, making it an efficient way of constructing a useful resource for standardization. This lexicon makes no attempt to model ambiguity, as a tweet wordform can sometimes have more than one standardization. We simply took the most obvious standardization candidate, typically without inspecting the corpus, which would have taken much more time. Sometimes one word is standardized to several standard words, i.e., a word is mapped to a phrase, so the relation between tokens in tweets and standardized ones is not necessarily one-to-one. Along with manual standardization, words were also flagged as being proper nouns (names), foreign words or errors in tokenization. The first are important as they can be OOV words as regards Sloleks, even though they are in fact standard words, the

[4] http://www-01.sil.org/linguistics/wordlists/english/

[5] http://wiki.apertium.org/wiki/Bosnian-Croatian-Montenegrin-Serbian_and_Slovenian

second as they are not really the subject of standardization, and the third as an error had been made in up-stream processing, so there is not much point in trying to standardize them. In this way we obtained a lexicon of 1,000 (195 of these flagged) of the most salient tweet-specific wordforms together with their standardized wordform, which constitutes part of the distributed dataset; we henceforth refer to this lexicon as the Training Lexicon, TL.

3.3 Manually annotated tweets

For development and testing various approaches we needed a collection of manually annotated tweets with typical Twitterese. We first filtered the vertical corpus file to select only interesting tweets, i.e., discarding those that are written in standard Slovene or have few Slovene words. The filter chooses tweets that have some Slovene words, less than half English words, more Slovene than Croatian words (note that each word can belong to more than one language), and at least a fifth of OOV words. We then took a sample of 10,000 lines from this collection and manually standardized and lemmatized it (the lemmatization was done in order to be able to use perform extrinsic evaluation of our standardization approach on this task as will be explained in Section 4.5). In the process of annotation, certain uninteresting Tweets, in particular the remaining ones in standard or foreign language, were discarded.

This gave us a manually corrected corpus of about 500 tweets and 7.500 tokens. The corpus was then split, one half to serve as the development set (TWEET-DEV), and the other as the test set (TWEET-TEST), both of which are part of the distributed dataset. The non-annotated remainder of tweets from our corpus was used to construct a resource for language modeling and CSMT hypothesis filtering containing in-vocabulary (IV) tokens with frequency higher than 10 only. We refer to this resource as TWEET-IV.

4 Experiments and results

Our overall approach to tweet standardization is based on standardizing only OOV tokens by applying transformations on them with the goal of producing wordforms identical to the ones produced during manual corpus standardization. Therefore we evaluate our approaches with two types of accuracy on the corpus:

1. ACC-ALL – accuracy on all word tokens in the corpus
2. ACC-OOV – accuracy on OOV word tokens in the corpus

The first measure reports how well we do on the level of complete texts, and the second one how well we do on the tokens we perform our transformations on.We perform all together five sets of experiments.

4.1 CSMT datasets

The first set of experiments attempts to identify the best subset of our TL lexicon for building the character-level translation model and the best target-language dataset for the character-level language model, along with the order of that language model. We perform evaluation on the TWEET-DEV dataset.

We experiment with all TL entries (ALL) and with TL entries where the original and standardized forms are different (DIFF). The results in Table 1 show that using all entries proves to be more informative than using just the entries where the original and standardized forms differ.

	ACC-ALL	ACC-OOV
ALL	**0.766**	**0.481**
DIFF	0.754	0.443

Table 1. Evaluation of the two TL subsets for building the translation model

Additional experiments with filtering the TL showed slight improvements when removing foreign words and errors in tokenization from the lexicon.

Regarding the order of the language models, we experiment with levels from 2 to 6. The best results are obtained with models of order 6, order 5 consistently producing slightly worse results, while lower-order LMs produce significantly worse results. We use Witten-Bell smoothing while constructing the language models.

We experiment with the following datasets for learning the character-level language model:

1. SLOLEKS – the inflectional lexicon of Slovene language
2. TWEET-IV-TOKEN – tokens from the non-annotated set of tweets with frequency above 10, confirmed in SLOLEKS
3. TWEET-IV-TYPE – types from the TWEET-IV-TOKEN dataset

The results in Table 2 show that significantly better results are obtained when using the TWEET-IV dataset than the SLOLEKS dataset which shows the benefits of using in-domain data. Using tokens rather than types, and thereby giving more probability mass to character sequences found in more frequent words improves the overall accuracy for 2.3 percent.

	ACC-ALL	ACC-OOV
SLOLEKS	0.720	0.335
TWEET-IV-TOKEN	**0.766**	**0.481**
TWEET-IV-TYPE	0.743	0.410

Table 2. Evaluation of different datasets for the character-level language model

4.2 Lower and upper bounds

The second set of experiments sets the lower (baseline) and upper bound (best possible performance, given the starting assumptions) of the remaining experiments calculated on the TWEET-DEV dataset.

We define two lower bounds, LB as the accuracy obtained without any intervention in the data while the second one, LB-TOP1, is the result of using the first hypothesis of the CSMT system obtained with the best performing settings from the first set of experiments.

We measure various upper bounds by inspecting n-best hypotheses from the CSMT system. We calculate UB-TOP5, UB-TOP10, UB-TOP20 and UB-TOP50. We calculate an overall upper bound UB as the accuracy if all OOV tokens were correctly standardized. Note that our method only standardizes OOV tokens and so cannot give perfect results, as some IV tokens are sort-of-false-friends between standard and non-standard language.

The results of calculating the lower and upper bounds are presented in Table 3. The LB lower bound shows that 26.6% of all tokens and 62% of OOV tokens require standardization. The LB-TOP1 lower bound, which applies the first CSMT hypothesis on OOV tokens, improves the overall accuracy by 3.2 points and OOV accuracy by 10.1 points.

The upper bounds calculated by taking into account n-best CSMT hypotheses show that most of the remaining correct hypotheses are positioned very high. We get an improvement of 5.4 points when taking into account the next four hypotheses and 3.8 points when inspecting the remaining 45 hypotheses.

The overall upper bound UB shows that the maximum overall accuracy, if all OOV tokens are standardized correctly, is 93.1%. There is a 7.3% gap between the UB-TOP50 and UB upper bound showing that the CSMT approach performs quite well (the difference between LB and UB-TOP50 is 12.4%), but that there is still room for improvement, probably by constructing a larger lexicon, i.e., producing more parallel data. There are 6.9% of tokens (1−UB) that are IV but require standardization, showing that future effort will have to be made in identifying and standardizing those tokens as well.

	ACC-ALL	ACC-OOV
LB	0.734	0.380
LB-TOP1	0.766	0.481
UB-TOP5	0.820	0.651
UB-TOP10	0.838	0.707
UB-TOP20	0.848	0.739
UB-TOP50	0.858	0.770
UB	0.931	1.0

Table 3. Different lower and upper bounds on TWEET-DEV

4.3 CSMT extensions

In the third set of experiments we compare the results of applying the TL only with different extensions of the basic CSMT approach on the TWEET-DEV dataset:

1. LEXICON – applying the TL only
2. CSMT-TOP1 – using the first hypothesis from the CSMT system (identical to the LB-TOP1)
3. CSMT-FILTER – using the first CSMT hypothesis if confirmed in the TWEET-IV dataset
4. CSMT-TOP5-FILTER –using the first of top 5 hypotheses confirmed in the TWEET-IV dataset
5. LEXICON-CSMT-FILTER – applying the TL and using the CSMT system with the TWEET-IV hypothesis filter as fallback for wordforms not covered in TL

The results of this set of experiments are presented in Table 4. Applying the TL only (LEXICON) performs significantly better than applying the first hypothesis of CSMT (CSMT-TOP1). By taking the first CSMT hypothesis only if confirmed in the TWEET-IV dataset (CSMT-FILTER), the CSMT approach does outperform the LEXICON approach with a small increase in accuracy of less than one point on all tokens and by 1.5 points on OOV tokens. Although LB-TOP1 and UB-TOP5 show that among hypotheses on positions 2-5 for 5.4% of tokens correct standardized wordforms can be found, choosing among the top 5 hypotheses the first one confirmed in the IV filter (CSMT-TOP5-FILTER) does outperform CSMT-TOP1, but underperforms regarding the simpler CSMT-FILTER. A possible explanation could be the fact that we work with a highly inflected language and that producing CSMT hypotheses covered by IV filters, but with wrong endings is pretty easy.

When combining the lexicon and the CSMT approach by using CSMT with the hypothesis filter as fallback on tokens not covered in the lexicon, we obtain the best results that outperform the LEXICON approach by 1.7 points on all tokens and 5.6 points on OOV tokens. With this joint setting we obtain an overall accuracy improvement on the TWEET-DEV dataset of 9.9 points on the whole corpus and 31.3 points on OOV tokens.

	ACC-ALL	ACC-OOV
LEXICON	0.816	0.637
CSMT-TOP1	0.766	0.481
CSMT-FILTER	0.820	0.652
CSMT-TOP5-FILTER	0.789	0.554
LEXICON-CSMT-FILTER	**0.833**	**0.693**

Table 4. Evaluation of various approaches to standardizing OOV tokens on TWEET-DEV

We performed additional experiments with using token-level LMs for reweighting CSMT hypotheses as performed in [9], but without any accuracy improvement. The probable reason is that we are already quite near the CSMT upper bounds calculated in the previous set of experiments. Namely, with the LEXICON-CSMT-FILTER setting we already obtained 80% of the maximum possible improvement in accuracy regarding the 50 best hypotheses produced by CSMT.

4.4 Lexicon vs. corpus standardization

The fourth set of experiments compares our lexicon approach of data preparation (LEX) to the standard approach of standardizing running text (COR) as performed in [9]. While all previous sets of experiments were evaluated on the TWEET-DEV dataset, here we use the TWEET-DEV dataset for building the lexicon from running text (the COR approach) and test both the LEX and COR approach on the TWEET-TEST dataset. We also consider this evaluation as final intrinsic evaluation of the LEXICON-CSMT-FILTER procedure constructed on the development set in the first three sets of experiments.

We construct the COR lexicon by taking a comparable amount of pairs of original and standardized forms from the TWEET-DEV dataset to the amount of forms in the TL by counting each entry where the original and standardized forms are identical as 0.5 and each entry where the forms differ as 1. We consider this to be a good estimate of the amount of effort necessary to inspect and possibly standardize each token in both approaches.

We present the results of the CSMT-FILTER and the LEXICON-CSMT-FILTER settings on both approaches in Table 5. We report the lower bound LB again, now calculated on the TWEET-TEST dataset.

The results show that in both approaches the lexicon approach (LEX) outperforms the corpus approach (COR). While the difference when using CSMT-FILTER is below one point, it does get more substantial when combining the lexicon and CSMT.

Comparing the CSMT-only and the joint approach of the two data annotation approaches we observe that, as one would expect, bigger improvement with the joint approach is achieved through the lexicon approach (1.5 points) than through the corpus approach (0.4 points). Nevertheless, using the deterministic approach for exact matches from the training corpus yields improvements on the corpus approach as well and should therefore be practiced.

It is important to note that, when constructing the lexicon from the corpus for the LEXICON-CSMT-FILTER approach, for each original form the most frequent (original form, standardized form) pair is used. The CSMT system is trained on all entries for an original form.

Last but not least, we compute the learning curves for both approaches as depicted in Figure 1. The left figure shows the results on using the CSMT setting with the TWEET-IV filter while the right figure shows the setting which uses filtered CSMT as fallback for the lexicon. The curves show that the LEX approach outperforms the COR approach on all sizes of the training data with

	ACC-ALL	ACC-OOV
LB	0.750	0.430
LEX-CSMT-FILTER	**0.841**	**0.716**
COR-CSMT-FILTER	0.836	0.701
LEX-LEXICON-CSMT-FILTER	**0.856**	**0.763**
COR-LEXICON-CSMT-FILTER	0.840	0.707

Table 5. Comparison of lexicon standardization to corpus standardization on TWEET-TEST

the COR approach slowly catching up as the training set size increases, more significantly in the CSMT-FILTER setting. All learning curves show room for additional improvement by annotating more data.

Annotating just one tenth of the data (100 tokens, below 20 minutes of annotation work) with the LEX approach and applying filtered CSMT already produces a significant improvement of 6.9 points to the LB lower bound which comprises 66% of the overall improvement obtained by the best performing setting of 10.4 points. The difference in accuracy between the LEX and the COR approach at that point is 4.1 points.

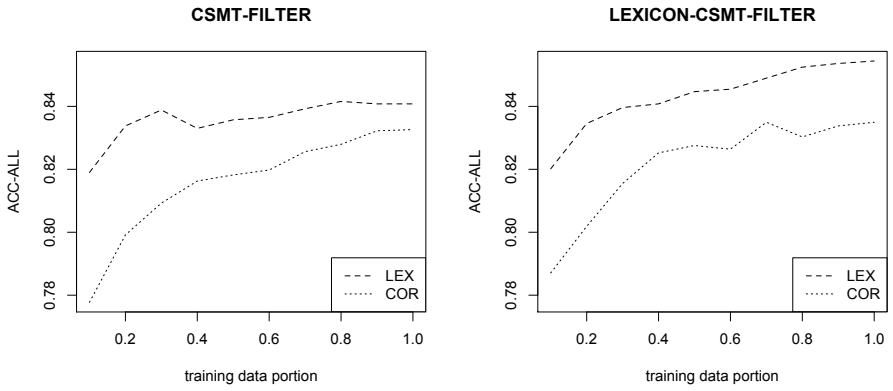

Fig. 1. Learning curves for the LEX and COR approach to data annotation

4.5 Lemmatization experiment

In order to extrinsically test the effect of our best scoring standardization, we performed a small experiment on a basic but very important task, at least for languages with rich inflectional morphology, namely lemmatization. Lemmatization abstracts away from inflectional variation and is useful for full-text search and dictionary lookup but is at the same time quite complex since Slovene words exhibit a complicated system of endings and stem alternations, dependent on morphological and syntactic features of the word, which makes learning Slovene inflections one of the more daunting tasks for foreign speakers.

Lemmatization was performed with ToTrTaLe [5], a program that tokenizes, transcribes, PoS tags and lemmatizes a Slovene text. The transcription module was developed to standardize historical Slovene texts and uses hand-constructed transcription patterns for this task. For the current experiment we either removed this step (in order to determine how well the system works with the original wordforms) or substituted it with our module for standardization. It should be noted that the PoS tagging step already receives the standardized wordforms, just as lemmatization, and that lemmatization makes use of the PoS tags because it is impossible to determine the lemma of a (at least OOV) Slovene word without it.

Table 6 shows the results of the experiment. The accuracy of lemmatization directly on "raw" words is just over 75%, while lemmatization accuracy on manually (i.e., perfectly) standardized words is almost 92%, so twitterese does indeed have a significant impact on processing of such texts. With automatically standardized words, the accuracy is almost 83.6%, which, as the last two columns show, is 8.5% better than on original data and just about equally worse than with perfect standardization, which should be taken as the upper accuracy bound we could achieve with the standardization. In other words, automatically standardizing the words cuts the absolute error rate by half.

RAW	AUTO	MANUAL	AUTO - RAW	AUTO - MAN
0.750	**0.836**	0.919	0.085	-0.083

Table 6. Comparison of lemmatization accuracy on original, manually standardized and automatically standardized wordforms from TWEET-TEST

5 Conclusions

In this paper we have presented a method for standardizing non-standard text, more specifically Slovene tweets, by using character-level SMT as fallback to lexicon lookup.

We compared the approach of manually standardizing most salient OOV tokens with respect to a reference corpus to the approach of standardizing running text. We have shown that the former produces significantly better results, especially on small training sets. This is an interesting finding given that we work with a highly inflected language with many possible forms that heavily depend on the context. For character-level language models we have shown that in-domain data performs better and that deduplication of tokens should not be performed. In both approaches to producing training data, using perfect match sequences from the parallel data, ie. performing lexicon lookup, and using CSMT only where there is no perfect match, showed to produce best results.

Filtering the first CSMT hypothesis with an in-vocabulary filter proved to be more useful than filtering the top 5, regardless of the fact that many correct hypotheses can be found on those positions. High flectiveness of the language

of interest and the danger of producing tokens with different endings covered in the IV filter is one possible explanation.

Finally, with our standardization approach we have shown that lemmatization errors produced on non-standard language are cut by half.

Regarding our future work, our primary goal is to extend our approach to more languages. We additionally plan on investigating a CSMT approach to standardization not limited to tokens, but applied on a wider context. By doing so we hope to deal with the 6.9% of tokens that are IV, but require standardization.

References

1. Špela Arhar: Učni korpus SSJ in leksikon besednih oblik za slovenščino. Jezik in slovstvo 54(3–4), 43—-56 (2009)
2. Aw, A., Zhang, M., Xiao, J., Su, J.: A Phrase-based Statistical Model for SMS Text Normalization. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions. pp. 33–40. COLING-ACL '06, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)
3. Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., Basu, A.: Investigation and Modeling of the Structure of Texting Language. Int. J. Doc. Anal. Recognit. 10(3), 157–174 (Dec 2007)
4. De Clercq, O.e., Desmet, B., Schulz, S., Lefever, E., Hoste, V.: Normalization of Dutch user-generated content. In: Proceedings of Recent Advances in Natural Language Processing. pp. 179–188. INCOMA (2013)
5. Erjavec, T.: Automatic linguistic annotation of historical language: ToTrTaLe and XIX century Slovene. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 33–38. Association for Computational Linguistics, Portland, OR, USA (June 2011)
6. Han, B., Cook, P., Baldwin, T.: Lexical Normalization for Social Media Text. ACM Trans. Intell. Syst. Technol. 4(1), 5:1–5:27 (Feb 2013)
7. Kaufmann, M., Kalita, J.: Syntactic Normalization of Twitter Messages. In: In Proceedings of the 8th International Conference on Natural Language Processing. ICON '10 (2010)
8. Pennell, D., Liu, Y.: Toward text message normalization: Modeling abbreviation generation. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. pp. 5364–5367 (2011)
9. Pennell, D., Liu, Y.: A Character-Level Machine Translation Approach for Normalization of SMS Abbreviations. In: Proceedings of 5th International Joint Conference on Natural Language Processing. pp. 974–982. Asian Federation of Natural Language Processing, Chiang Mai, Thailand (November 2011)
10. Rayson, P., Garside, R.: Comparing Corpora Using Frequency Profiling. In: Proceedings of the Workshop on Comparing Corpora - Volume 9. pp. 1–6. WCC '00, Association for Computational Linguistics, Stroudsburg, PA, USA (2000)
11. Scherrer, Y., Erjavec, T.: Modernizing historical Slovene words with character-based SMT. In: BSNLP 2013 - 4th Biennial Workshop on Balto-Slavic Natural Language Processing. Sofia, Bulgarie (Jul 2013)