Rule-Based Normalization of German Twitter Messages

Uladzimir Sidarenka, Tatjana Scheffler, and Manfred Stede

University of Potsdam,
Karl-Liebknecht Str. 24-25
14476 Potsdam
{uladzimir.sidarenka,tatjana.scheffler,manfred.stede}@uni-potsdam.de

Abstract. In this article, we conduct quantitative and qualitative analyses of unknown words in German Twitter messages, and propose a normalization method which prepares German tweets for standard text processing tools. In the first part, the prevalence of different types of out-of-vocabulary (OOV) tokens and non-standard language in German Twitter data is determined. In a second step, we present a set of ad-hoc techniques which can tackle some of the most prominent effects found during the analyses. We show how this set of techniques helps us lower the average rate of out-of-vocabulary tokens in Twitter messages and how this lower OOV-rate in turn helps improve the quality of automatic part-of-speech tagging.

Keywords: twitter, social media, text normalization, spelling correction

1 Introduction

When Jack Dorsey, the present CEO of Twitter Inc., was sending the very first tweet on March 21, 2006 (Dorsey, 2006), he probably did not realize that his message – "just setting up my twttr" – already contained a word which was unknown to the majority of NLP applications existing at that time and that there would be many such words in tweets in the future causing a lot of problems for automatic text analysis tools.

And though the problem of out-of-vocabulary words and textual normalization have been extensively studied in computational linguistics since as early as the late 1950s (cf. Petersen, 1980) and were anything but new at the time when online communication emerged, it were small messages that revived interest in these fields in the past two decades.

In the next section, we will give an overview of existing scientific approaches to the problem of tackling text noisiness in non-standard texts. After that, in Section 3, we will analyze which types of noisiness phenomena are especially characteristic for German Twitter. Section 4 will subsequently describe an automatic procedure for mitigating some of the most prominent of those effects. In a concluding step, we will perform an evaluation of the results of this procedure and give some possible suggestions for future research.

2 Related Work

At the very onset of the works on noisy text normalization (NTN), the two major sources of data where most text noisiness came from were texts produced by automatic optical character recognition and speech recognition applications. A relatively low average accuracy of those applications at the beginning of the 1990-s forced researchers to think about how words distorted during recognition could be restored to their respective standard language forms at the end of the processing. A method which seemed to be most suitable for these purposes at that time was the technique called "noisy channel model" (NCM) first proposed by Shannon in 1948.

NCM divided the complex task of text normalization in three smaller subproblems which could be formulated as follows *a*) given an unknown and purportedly not normalized word (NNW) how could its normalized variants (NVs) be retrieved; *b*) given the devised NVs, how probable would it be that they really were the correct variants for normalizing NNW; and finally *c*) given the devised NVs, how probable would it that they would ever occur in a normalized text.

It is no wonder that NCM was one among the first methods which was applied for normalization of mobile messages and Internet-based communication (IBC) texts. In 2003, Clark proposed a unified NCM-based system which jointly performed tokenization, sentence splitting, and word normalization of Usenet forum posts. Choudhury et al. (2007) extended the NCM-approach proposed by Toutanova and Moore (2002) by converting it to a Hidden Markov Model and then applied this model for normalization of SMS messages. Beaufort et al. (2010) made use of finite state methods (FSM) to perform French SMS normalisation by combining the advantages of FSM and NCM.

Starting from the early 2000-s, the increasing quality of statistical machine-translation (SMT) applications and gradual realization of the shortcomings of NCM methods operating solely on the character level of words spurred the researchers on the development of NTN methods which were more similar to the established SMT techniques. In the scope this framework, normalization task was defined as a task of mapping an unnormalized *word phrase* to its normalized counterpart. This counterpart could either be specified manually as was done by Clark and Araki (2011) or it could be derived automatically during the alignment of normalized and unnormalized training data as was suggested by Aw et al. (2006). The latter approach however presupposed that a sufficiently large collection of such data was available. One of the first attempts to apply an SMT-like technique to normalization of Twitter messages was made by Kaufmann (2010) who used a corpus of SMS messages as his training set though.

Finally, as it was noticed by Kobus et al. (2008), NTN methods relying on either NCM or SMT techniques usually revealed complementary strengths and weaknesses. This notion led to the idea that incorporating these two normalization approaches into one system would improve the overall performance as different sources of information would benefit from each other. So Kobus et al. (2008) proposed an approach which first used a trained SMT module and then fed its

output into a finite state transducer (FST) whose transitions represented phonetic or graphematic substitutions frequently occurring in unnormalized words.

It should however be noted that almost all of the above methods mainly concentrated on only English data. A few exceptions from this were approaches suggested by Beaufort et al. (2010) and Kobus (2008) for French. Oliva et al. (2013) proposed a hybrid procedure for normalization Spanish SMSes. To the best of our knowledge, not very much research has been done for German in this field so far. In order to get a better intuition what the nature of words requiring normalization is in this language and to be able to answer the three fundamental questions formulated by NCM, we decided to have a closer look at German Twitter and to analyze words occurring there which were regarded as unknown by standard NLP tools commonly used for processing German texts.

3 Analysis of Unknown Tokens

In order to estimate the percentage of unknown words in Twitter, we randomly selected 10,000 messages from a corpus of 24,179,871 German tweets gathered in April 2013. We developed a Twitter-aware sentence splitter. For tokenization, we slightly adjusted the specialized Twitter tokenizer[1] developed by Christopher Potts to the peculiarities of German. After skipping all words which did not contain any alphabetic characters (i.e., numbers and punctuation marks) or consisted only of a single letter, we obtained a list of 129,146 tokens. As reference systems for dictionary lookup we used the open-source spell checking program `hunspell`[2] and the publicly available part-of-speech tagger `TreeTagger`[3] (Schmid, 1994).

Out of this token list, 26,018 tokens (20.15 %) were regarded as unknown by `hunspell` and 28,389 tokens (21.98 %) were considered as OOV by `TreeTagger`. We also performed these estimations for word types. The relative rate of unknown words raised as expected and ran up to 46.96 % for `hunspell` and 58.24 % for `TreeTagger`, out of a total of 32,538 types.

We classified found OOV tokens into the following three groups according to the reasons why these tokens could have been omitted from applications' dictionaries:

1. **Objective limitation of machine-readable dictionaries (MRD)**. Among this group, we counted valid words of basic vocabulary which had erroneously been omitted from an applications' MRD;
2. **Stylistic specifics of text genre**. This group comprised words which did not belong to the standard language but were perfectly valid terms in the domain of web discourse or more specifically in Twitter communication;
3. **Spelling deviations**. In the scope of this group, we considered non-standard spellings of words encountered in text.

[1] `http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py`
[2] Ispell Version 3.2.06 (Hunspell Version 1.3.2); dictionary de_DE.
[3] Version 3.2 with German parameter file UTF-8.

In order to see how detected OOV words were distributed among and within these 3 major groups, we manually analyzed all OOV tokens which appeared in the text more than once and also looked at 1,000 randomly selected hapax legomena. The breakdown of these OOV tokens into the three major classes is shown in Table 1. More fine-grained analysis is discussed below.

Table 1. Distribution of OOV words over the three major classes

OOV subclass	`hunspell`		`TreeTagger`	
	% of OOV tokens	% of OOV types	% of OOV tokens	% of OOV types
Objective limitation of MRD	45.87	54.62	40.46	43.36
Stylistic specifics of text genre	41.65	33.64	48.02	44.59
Spelling deviations	11.87	10.75	9.09	8.23
Intended deviations	8.06	5.09	5.97	3.7
Unintended deviations	3.81	5.66	3.12	4.54

We subdivided the class 1 into the following subcategories:

1. regular German words, e.g. *Piraterie*, *losziehen*;
2. compounds, e.g. *Altwein*, *Amtsapothekerin*;
3. abbreviations, e.g. *NBG*, *OL*;
4. interjections, e.g. *aja*, *haha*;
5. named entities, with subclasses:
   (a) persons, e.g. *Ahmadinedschad*, *Schweiger*;
   (b) geographic locations, e.g. *Biel*, *Limmat*;
   (c) companies, e.g. *Apple*, *Facebook*;
   (d) product names, e.g. *iPhone*, *MacBook*;
6. neologisms, with subclasses:
   (a) newly coined German terms, e.g. *entfolgen*, *gegoogelt*;
   (b) loanwords, e.g. *Community*, *Stream*;
7. and, finally, foreign words like *is* or *now* which in contrast to 6b were not mentioned in any existing German lexica and did not comply with inflectional rules of German grammar.

Such taxonomy was supposed to reflect the fact that valid words could have been omitted from an MRD either due to the limitations of developers' capacities (group 1), active word formation processes or lexical productivity of the language itself (groups 2 through 6a) or also due to language's openness to foreign language systems (groups 6b and 7).

Percentage figures for each of the above subgroups are shown in Table 2. We have considered OOV-distributions for both `hunspell` and `TreeTagger`. For each of them, we estimated the percentage of a particular subclass with regard

to the total number of occurrences of all OOVs (column "% of OOV tokens") as well as with regard to the percentage rate in the list of only unique unknown tokens disregarding their frequencies (column "% of OOV types").

Table 2. Distribution of OOV words belonging to the class "Objective limitedness of MRD"

OOV subclass	`hunspell`		`TreeTagger`	
	% of OOV tokens	% of OOV types	% of OOV tokens	% of OOV types
regular German words	7.27	8.66	2.74	3.46
compounds	1.27	2.65	2.5	4.54
abbreviations	3.96	4.8	3.26	3.43
interjections	5.99	4.6	5.56	4.27
person names	4.77	6.49	2.31	3.46
geographic locations	1.53	2.6	1.16	1.87
company names	2.28	2.87	4.34	3
product names	2.16	2.65	2.45	3.22
newly coined terms	1.37	1.35	3.32	2.38
loanwords	3.7	4.06	3.28	2.86
foreign words	11.57	13.89	9.54	10.87

Similarly to class 1, we divided the group – "Stylistic specifics of text genre" – into the following subclasses:

1. @-tokens, e.g. *@ZDFonline*, *@sechsdreinuller*;
2. hashtags, e.g. *#Kleinanzeigen*, *#wetter*;
3. links, e.g. *http://t.co*, *sueddeutsche.de*;
4. smileys, e.g. *:-P*, *xD*;
5. slang, e.g. *OMG*, *WTF* etc.

according to the formal or lexical class which tokens of this group belonged to. As slang, we considered colloquial and dialectal lexical expressions (e.g. *nö*, *bissl*), common phrases pertaining to the genre of internet-based communication (e.g. *LOL*, *ava*), as well as spellings of words which resembled their colloquial pronunciation in everyday speech (e.g. *Tach* instead of *Tag*, *grade* instead of *gerade*). The latter cases were assigned by us to two categories, namely, *slang* and *spelling deviations*. Detailed statistics on the afore-mentioned subgroups of class 2 are shown in Table 3:

A striking outlier of 16.22 % for slang tokens in column 1 of the Table is explained by the fact that the word "RT" which occurred 1,235 times in our texts and was by far the most frequent OOV in the analyzed data set, was recognized as OOV by `hunspell` but was not deemed as such by `TreeTagger`.

The last major class is "Spelling deviations", which includes both intended (see above for "slang") and unintended spelling variations. It was split into the groups:

Table 3. Distribution of OOV words belonging to the class "Stylistic specifics of text genre"

OOV subclass	`hunspell`		`TreeTagger`	
	% of OOV tokens	% of OOV types	% of OOV tokens	% of OOV types
@-tokens	13.12	20.49	16.14	21.84
hashtags	7.41	6.26	13.02	10.56
links	2.45	0.4	4.88	6.05
smileys	2.01	0.74	6.86	1.2
slang	16.66	5.75	7.12	4.94

1. insertions, e.g. *dennen* instead of *denen*;
2. deletions, e.g. *scho* instead of *schon*;
3. substitutions, e.g. *fur* instead of *für*;

according to the type of operation which led to a particular spelling mistake. In cases when multiple different operations were involved simultaneously on one word, we explicitly marked each of these operations in our data. Statistical distribution of these subclasses is shown in Table 4 on page 6.

Table 4. Distribution of OOV words belonging to the class "Spelling deviations"

OOV subclass	`hunspell`		`TreeTagger`	
	% of OOV tokens	% of OOV types	% of OOV tokens	% of OOV types
insertions	1	1.66	0.79	1.08
deletions	8.3	6.28	6.55	5.33
substitutions	2.57	2.81	1.75	1.82

As is clear from the table, deletions are by far the most common type of deviant spellings. This is partially explained by either deliberate or accidental omissions of characters made by users, but an even bigger part of deletions was due to the automatic truncations of too long messages which were performed by the Twitter service itself.[4]

Since the latter two major groups of OOVs (Twitter-specific phenomena and spelling deviations) accounted for more than a half of all unknown tokens found in Twitter and posed a significant problem for automatic analysis, we decided to address these classes by applying a set of normalization procedures to each of them.

[4] Since Twitter imposes a strict restriction of 140 characters on the length of posted messages, longer tweets get automatically truncated.

4 Text Normalization Procedure

4.1 Replacement of Twitter-Specific Phenomena

In order to remove noise caused by Twitter-specific phenomena (TSP) and to reduce data sparsity for further processing, we replaced TSPs, which played a significant syntactic role in a sentence, with generic tokens representing the class of TSP being replaced. Those phenomena which did not have any particular syntactic function and did not bear any relevant semantic information were deleted from messages. This approach is similar to the *syntactic disambiguation* steps suggested by Kaufmann (2010) for normalization of English tweets.

For our purposes, we developed a prototypic Python system analogous to a finite-state transducer in which a set of regular expressions was associated with corresponding actions performed on matched subgroups.

In this system, we replaced all smileys with the tokens "%PosSmiley" or "%NegSmiley", depending on the type of emotion conveyed by a particular emoticon. In cases when the polarity of a smiley was unclear, the generic substitution token "%Smiley" was used. For hyperlinks and e-mail addresses, we looked at the surrounding context. If these items occurred outside of any sentence, and were not preceded or followed by a preposition or conjunction, we removed them from the text. Otherwise, these tokens were replaced with the dummy words "%Link" and "%Mail" respectively. Furthermore, we stripped all leading "#" characters from the beginning of hashtags, since the alphabetic part of these tokens practically always bore some significant semantic information, even if these tags appeared outside of a sentence.

For @-mentions, we had to decide whether the occurrences of these mentions were meant simply as reply addresses or formed an indispensable constituent in a sentence. In the former case, @-mentions were deleted, in the latter case, we replaced them with the artifical token "%Username".

When multiple rules matched the same context, we decided which rule to apply by looking at the starting and ending positions of the first matched subgroups of each of the matched expressions, the starting and ending positions of the whole regular expressions, the number of subgroups in each generated match object and, finally, the order in which these rules appeared in our rule file.

We added all introduced artificial tokens to the custom dictionaries of `TreeTagger` and `hunspell`. Furthermore, we remembered positions and lengths of all made replacements along with the original input words which were deleted or replaced, so that a restoration step could be performed any time after the processing.

4.2 Restoration of spelling deviations

In order to get a deeper insight into the nature of incorrect spellings in Twitter, we calculated what part of spelling deviations in our annotated data were also assigned to the *slang* category. Such *slang* or *colloquial* spelling variants accounted for 67,06 % of all spelling deviations found by `hunspell` and formed

64,36 % of all spelling deviations detected by `TreeTagger`. Another noticeable fact about these spellings was that the prevailing majority of them occurred more than once in our texts. This could be explained by the fact that colloquial spelling variants are usually formed by more or less regular processes. Such processes are commonly applied to frequently used words which appear over and over again in text. Non-colloquial misspellings, on the contrary, are formed by occasional slips of finger. So, users neither notice nor tend to repeat them later.

According to our data, the most productive processes which produced most of the colloquial spelling variants were:

- Omission of 'e' in unstressed positions, e.g. *würd*, *zuguckn* etc. In cases when 'e' was part of the impersonal pronoun "es" following a verb, the remaining 's' of this pronoun was usually appended to the preceding verb form, e.g. *wirds* instead of *wird es*;
- Omission or replacement of final consonants with their voiceless equivalents, e.g. *nich* instead of *nicht* or *Tach* instead of *Tag*;
- Multiple repetitions of characters as a means of expressing elongation of sounds, e.g. *Hilfeeee*, *süüüß*;
- Omissions of 'ei' from indefinite articles, e.g. *ne* instead of *eine* or *nem* in lieu of *einem*;
- Omissions of 'he' from the verbal prefixes *herauf-*, *heraus-*, *herum-* etc., e.g. *rauszukriegen*, *rumbasteln*, ;

We developed a set of reverse transformation rules which first captured tokens with suspicious character sequences, then checked whether the captured word was not present in the dictionary and whether its assumed transformation was a valid in-vocabulary term. If these conditions were satisfied, we applied the transformation associated with this rule. We have tested our set of 11 rules on a held-out corpus of 184,331 tweets. It turned out, however, that our dictionary checks were insufficient though, since they did not prevent us from making such incorrect changes like the one shown in the Example 1.

Example 1. Wulff tritt zurück, Georg **Schramme** wird neuer Bundespräsident

In this sentence, the last name "Schramm" was incorrectly replaced with the word "Schramme", since the former token was unknown to the dictionary whereas the latter was found in vocabulary. To prevent such erroneous corrections, we decided to incorporate statistical information into our system and added the restriction to the rules producing errors, that for a given unknown input word w_i and its possible in-vocabulary suggestion w_i^*, the following inequality had to be satisfied:

$$\begin{aligned} log(P(w_{i-1}, w_i)) + log(P(w_i)) + log(P(w_i, w_{i+1})) < \\ log(P(w_{i-1}, w_i^*)) + log(P(w_i^*)) + log(P(w_i^*, w_{i+1})) \end{aligned} \tag{1}$$

This inequality means that the sum of log-probabilities of a preceding bigram, current unigram and the immediately following bigram for the word to be replaced had to be lesser than the corresponding sum of log-probabilities for its

possible substitution.[5] We gathered both the unigram and the bigram statistics from our held-out set of tweets and smoothed them using add-λ smoothing. This statistical information was later also used in the evaluation phase.

A different technique was used for tackling elongations of characters. For these, we first squeezed successive repetitions of more than three characters in a row to a maximum of three repetitive characters. After that, all possible combinations of consecutively squeezed repeated characters were generated. It means that for a word like "daaaaaassss", we first transformed it to "daaasss" and then generated all possible variants with triple, double and single repetitions of "a" and "s".

From this generated set, we removed all candidates which did not appear in the dictionary. If multiple candidates were left after that, e.g., "dass" and "das", these candidates were scored using the joined sum of bigram and unigram log-probabilities as described in the inequality 1. If no word in the generated set was found in the reference dictionary, we fell back to the method suggested by Brody and Diakopoulos (2011) and replaced each repeated letter with a single instance of that letter. For each such condensed form, there was a mapping to the most frequently elongated form occurring in a training corpus from which this dictionary was generated. If no condensed form was found, we simply returned its squeezed form with maximum consecutive repetitions of three characters.

A much harder case for normalization represented spelling variants which were classified as true, i.e. non-colloquial, misspellings. Such incorrect spellings did not show any regularities except for the cases of incorrect spelling of umlauts. The German characters *ä*, *ö*, and *ü* were often rewritten as sequences *ae*, *oe*, and *ue* respectively. Since these character sequences also often appear in common German words like, *Mauer*, *Feuer*, *virtuell*, we extracted all words having these character sequences from a corpus of newspaper articles which were assumed to be typed with correctly spelled umlauts. We replace the sequences *ae*, *oe*, and *ue* with their respective umlauts, only if they did not appear as correctly spelled words in the newspaper corpus.

5 Evaluation

We performed both an intrinsic and an extrinsic evaluation of the effectiveness of our normalization procedure. As an intrinsic evaluation, we first measured how the relative rate of OOV words in the input text changed after normalization. It turned out that this rate decreased by 5.6 % for `hunspell` and by 8.9 % for `TreeTagger`. This significant decrease, however, was mainly caused by our replacement/removal of Twitter-specific terms and the addition of the artificial replacement tokens to the applicatons' dictionaries.

In order to separately assess the performance of our spelling correction module, we extracted all messages from our analyzed corpus of 10,000 tweets which

[5] One anonymous reviewer suggest to use a fixed threshold instead of only requiring the probability of the replacing token to be higher than the probability of the current token. We leave this interesting proposal for further research.

contained at least one word that was marked as a deviant spelling in the annotated data. We constructed hand-corrected gold data from this as follows: We automatically replaced the deviant spellings with their correct variants which we had previously specified during the annotation. However, since we did not annotate all OOVs which occurred in the text only once but only 1,000 randomly selected hapax legomena, we manually corrected unknown words which were not included in our annotation. After that, extracted messages were pre-processed by replacing Twitter-specific phenomena as described in Section 4.1. This gave us a test set of 1,492 messages in which 1,480 misspellings were to be corrected.

For comparability, we evaluated the performance of our system by computing the token-level precision, recall, and F-score ($\beta = 1$) of replacements. Additionally, we measured the BLEU and NIST scores between the gold standard versions and the unprocessed resp. the normalized tweets. The results of our evaluation are shown in Table 5.

Table 5. Evaluation results for the spelling correction module

Input text	BLEU	NIST	Precision	Recall	F-score
Before normalization	0.7929	12.55	–	–	–
After applying SDC	0.8455	12.9873	0.84	0.29	0.4317
After applying SDC + TSI	0.8638	13.1474	0.8750	0.383	0.5328
After applying SDC + TSI + ES	0.8687	13.1971	0.875	0.4162	0.5641
After applying SDC + RSE + ES + UR	0.8766	13.2638	0.8793	0.4584	0.6027

Finally, as a first step towards an extrinsic evaluation, we measured how the performance of POS-tagging changed after normalization. For this, we randomly picked 200 tweets from our analyzed data and POS-tagged them before and after normalization using `TreeTagger`. After manual inspection of the results, we could see a performance increase by 16.4 % from 71.82 % to 88.22 %.

6 Conclusions and Future Work

With this article, we hope to have provided a better insight into the nature of ill-formed words in German Twitter messages. As was shown in Section 3, special markup elements and casual spellings account for more than half of the unknown words discovered in tweets. Furthermore, almost three quarters of non-standard spellings could be regarded as colloquial spelling variants rather than occasional slips of the finger. Such colloquial spellings also showed the tendency to be formed by well formalized processes and to be used frequently in text.

We suggested a rule-based text normalization approach which could serve as a baseline comparison measure for future normalization methods which may be suggested for German tweets. As was shown in previous sections, our approach

can effectively address some of the most frequent phenomena which contribute to a higher rate of out-of-vocabulary words in Twitter texts, such as Twitter-specific elements and non-standard spellings.

We are going to make our classification and test data available online under the terms consistent with Twitter regulations. Our annotated data could, in our opinion, significantly reduce tedious manual work for other researchers.

In the future, our method could be further refined and improved. As possible steps for such refinement, we see on the one hand the addition of a machine-learning classifier which could help distinguish spelling mistakes from valid unknown words. On the other hand, a better disambiguation of suggested replacement variants for misspellings could be achieved by letting a part-of-speech tagger process a word grid with input tokens and their suggested replacements. At the end of tagging, not only the most probable tag sequence but the most likely word/tag mesh could be chosen in a single step. While these improvements are beyond the scope of this work, we hope to have laid the groundwork for them.

Acknowledgement.

This work was funded by the Federal Ministry of Education and Research as a part of the collaborative project "Analysis of Discourse in Social Media" (project number 01UG1232D). The authors would like to thank two anonymous reviewers for their helpful comments. We are also grateful to Martin Schwietzke for his help with the annotation of data.

References

Aw, A., Zhang, M., Xiao, J., Su, J.: A Phrase-based Statistical Model for SMS Text Normalization. COLING/ACL (2006) 33–40

Bangalore, S., Murdock, V., Riccardi, G.: Bootstrapping Bilingual Data using Consensus Translation for a Multilingual Instant Messaging System. COLING (2002) 33–40

Beaufort, R., Roekhaut, S., Cougnon, L. A., Fairon, C.: A Hybrid Rule/Model-Based Finite-State Framework for Normalizing SMS Messages. ACL (2010) 770–779

Brill, E., Moore, R. C.: An improved model for noisy channel spelling correction. ACL (2000) 286–293

Brody, S., Diakopoulos, N.: Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs In Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2011) 562–570

Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., Basu A.: Investigation and Modeling of the Structure of Texting Language. International Journal of Document Analysis and Retrieval: Special Issue on Analytics of Noisy Text. 10 (2007) 157–174

Church, K., Gale, W.: Probability scoring for spelling correction. Statistics and Computing. 1 (1991) 93–103

Clark, A.: Pre-processing very noisy text. In Proceedings of Workshop on Shallow Processing of Large Corpora. Corpus Linguistics. Lancaster. pp. 12–22 (2003)

Clark, E., Araki, K.: Text Normalization in Social Media: Progress, Problems and Applications for a Pre-processing System of Casual English. PACLING. Procedia - Social and Behavioral Sciences 27 (2011) 2–11

Cook, P., Stevenson, S.: An unsupervised model for text message normalization, Proceedings of the Workshop on Computational Approaches to Linguistic Creativity. CALC '09. (2009) 71–78

Dorsey, J.: "just setting up my twttr". `https://twitter.com/jack/status/20` Accessed February 26, 2013. (2006)

Kaufmann, M.: Syntactic normalization of twitter messages. The 8-th International Conference on Natural Language Processing. (2010)

Kobus, C., Yvon, F., Damnati, G.: Normalizing SMS: are Two Metaphors Better than One? COLING (2008) 441–448

Krawczyk S., Raghunathan, K.: Investigating sms text nor-malization using statistical machine translation. Stanford University, Stanford, CA, 2009

Mayes, E., F. Damerau, et al.: Context Based Spelling Correction. Information Processing and Management. 27(5) (1991) 517–522

Oliva, J., Serrano, J. I., and Del Castillo, M. D., and Igesias, .: A SMS normalization system integrating multiple grammatical resources. Natural Language Engineering. (2013) 121–141

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation. ACL (2002) 311–318

Petersen, L. J.: Computer Programs for Detecting and Correcting Spelling Errors. Communications of the ACM 23/ 12 (1980) 676–687

Pianigiani, G., Donadio, R.: Twitter Has A New User: The Pope. The New York Times. Page A6. (December 4, 2012)

Sproat, R., Black, A. W., Chen, S. F., Kumar, S., Ostendorf, M., Richards, Ch.: Normalization of non-standard words. Computer Speech & Language, 15/3 (2001) 287–333

Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proceedings of International Conference on New Methods in Language Processing. (1994)

Shannon, C. E.: A mathematical theory of communication. Bell system techical journal, 27:379–423 (1948) 623–656

Toutanova, K., Moore, R. C.: Pronunciation Modeling for Improved Spelling Correction. ACL (2002) 144–151