MT @HanBaldwin: Fightin OOVs in German #twitter

Uladzimir Sidarenka, Tatjana Scheffler, and Manfred Stede

University of Potsdam,
uladzimir.sidarenka@uni-potsdam.de, tatjana.scheffler@uni-potsdam.de,
manfred.stede@uni-potsdam.de,
WWW home page: http://www.ling.uni-potsdam.de/acl-lab/SocMedia/main.htm

Abstract. This article gives an overview of existing approaches to the problem of out-of-vocabulary (OOV) tokens and noisiness phenomena in natural language texts. These approaches are classified with regard to the size of text spans and knowledge inference mechanisms which they rely on in their work. Additionally to that, we conduct quantitative and qualitative analyses of unknown words in German Twitter messages, in order to see how relevant the OOV and text normalization problems are for this particular kind of micro-texts and what the characteristics of these OOVs are. In a concluding step, we present a set of ad-hoc techniques which are supposed to tackle some of the most prominent disturbing effects found during the analyses and 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 could not imagine that a few years later presidents and government officials would use this service to communicate with their voters and the Pope would be posting short messages holding an iPad in his hand (Pianigiani, 2012). Yet another thing that Jack Dorsey was apparently not aware of at that moment, was the fact 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 of such words in future causing a lot of headache to natural language specialists.

Though the problem of out-of-vocabulary words and its closely related task of textual normalization have been extensively studied in computational linguistics since as early as the late 1950s (cf. Petersen, 1980) and were certainly anything but new at the time when mobile communication emerged, it were small messages that revived interest in this research area in the past two decades.

In the next Section we will give a short overview of existing scientific approaches to the problem of tackling out-of-vocabulary words 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 further suggestions for future research.

2 Related Work

Before proceeding with the description of recent methods for noisy text normalization (NTN), we first would like to define the criteria by which these methods could be classified. It should be noted that there already exist several classifications of NTN approaches including, for example, Kukich (1992) and Kobus et al. (2008).

Kukich (1992), for instance, divided all NTN techniques into six classes:

1. minimum edit distance techniques;
2. similarity key techniques;
3. rule-based techniques;
4. n-gram based techniques;
5. probabilistic techniques;
6. neural nets.

Kobus (2008), on the contrary, referred to NTN methods as *metaphors* and split them into the following three groups:

1. "spell checking" metaphor;
2. "translation" metaphor;
3. "speech recognition" metaphor.

Though both of these divisions seem to be justified to some extent, it is difficult to determine using them whether an NTN approach that detects and restores incorrectly spelled words on the basis of phonetical n-gram statistics should fall into the n-gram based, probabilistic, spell checking or speech recognition class.

The reason for this confusion is the fact that the above classifications both rely on several independent criteria at the same time, though each of these criteria characterizes an NTN system from a different point of view. As a consequence, unambiguous assignment of an NTN approach to one particular class often becomes impossible. In order to avoid this, we instead suggest using separate classifications for each type of involved criteria which might characterize an NTN technique.

One of such criteria which in our opinion would be worth a separate classification is *segmentation level* that is used *a*) to infer in-vocabulary (IV) equivalents for OOV tokens and *b*) to choose the most probable variant among multiple

possible suggestions[1]. For this criterion, we propose division into the following classes:

1. graphematic[2];
2. lexical;
3. phrasal.

Each broader level of this hierarchy is supposed to either incorporate or ignore information provided by its narrower subsegments. In this way, we only need to mention one (the broadest) hierarchical class for the cases when multiple segmentation levels are involved by some techniques.

The second criterion regards the *type of information induction* that is used to devise the correction rules. This leads us to the usual NLP-taxonomy which divides all approaches into:

1. rule-based;
2. statistical[3];
3. and hybrid ones.

According to these two classification criteria, recent approaches to the NTN task could be grouped together as follows.

The earlier works on NTN commonly relied on either purely graphematic or phonographematic levels of segmentation for deriving normalization variants of incorrectly spelled words. To purely graphematic systems belong methods suggested by Brill and Moore (2000), Sproat et al. (2001), and Clark (2003). As phonographematic approaches one could regard works done by Toutanova and Moore (2002), Choudhury et al. (2007), Cook and Stevenson (2009), Han and Baldwin (2011) etc. With regard to the type of information inference, most of these methods were supervised with the exceptions of Cook and Stevenson (2009) and Han and Baldwin (2011) who claimed to use an unsupervised technique. Sproat et al. (2001) described in their article both a supervised and unsupervised approach.

Starting from the second half of the 2000s, the raising influence and improved quality of machine translation tools led to the development of NTN technologies which used broader levels of segmentation. In 2006, Aw et al. suggested a supervised statistical system for normalization of SMS-messages which operated on automatically aligned phrases A few years later, Clark and Araki (2011) described a purely rule-based method which used mappings of non-standard words and phrases to their corresponding standard language forms.

[1] For the cases, when segments of different lengths are used for tasks a and b, we note it explicitly in our classification on which segmentation length each of these subtasks relies.

[2] Depending on whether phonetical information is involved or not at this level, this class could be further divided into a phonographematic and purely graphematic subclasses.

[3] Depending on the type of training data required, this class is in turn usually divided into unsupervised, semi-supervised, and supervised groups.

As noted by Kobus et al. (2008), NTN methods relying on either graphematic or phrasal segments usually revealed complementary strengths and weaknesses. This notion led NLP scientists to the idea that by incorporating multiple levels of the language into one NTN system the total performace of the whole system would improve as different sources of information would benefit from each other. As a consequence of this, a wealth of combined techniques emerged in the past few years. Among these we should especially mention works by Kobus et al. (2008), Kaufmann (2010), and Oliva et al. (2013). The majority of these systems used the whole range of segmentation levels from phonographematatic to phrasal one, and in many cases they also applied different knowledge inference mechanisms to different levels of the language.

It should however be noted that almost all of the above methods mainly concentrated on only English data. A few exceptions from that are approaches suggested by Beaufort et al. (2010) for French, and Oliva et al. (2013) for Spanish. To find out which peculiarities of ill-formed words are characteristic for German, we will perform a quantitive and qualitative analysis of unknown words in German Twitter in the next Section in order to see what kind of NTN techniques would be most suitable for handling such words there.

3 Analysis of Unknown Tokens

In order to estimate the percentage of unknown words in Twitter messages, we randomly selected 10,000 tweets from a previously collected corpus, split them into sentences and tokenized using social media-aware tokenizer by Christopher Potts [4]. After skipping all words which did not contain any alphabetic characters or consisted only of a single letter, we obtained a list of 129,146 tokens. As reference systems for dictionary lookup we used open-source spell checking program `hunspell`[5] and publicly available part-of-speech tagger `TreeTagger`[6] (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 similar estimations after leaving only unique words without taking into account their frequencies. This allowed us to shrink our initial token list by four times to 32,538 unique tokens. The relative rate of unknown words raised as expected and run up to 46.96 % for `hunspell` and 58.24 % for `TreeTagger`.

Here once again the question of classification arose. This time with regard to the reasons why different tokens could have been omitted from corresponding applications' dictionaries. In this respect, division into following groups seemed to us to be appropriate to us:

1. **Objective limitedness of machine-readable dictionary (MRD)**. Among this group, we counted words of basic vocabulary which did not get into ap-

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

plications' MRD either because they supposedly were rare or because they did not exist at the time when dictionaries were created. Another reason for inclusion in this type was the belonging of a word to an open lexical or part-of-speech class (like, for example, named entities or interjections) which are often omitted from MRDs due to the impossibility to fully cover them;
2. **Sloppiness of users' input**. In the scope of this second group, we considered incorrect spellings of words encountered in text;
3. **Stylistic specifics of text genre**. This group comprised words which could be considered as illegal from the point of view of standard language but were perfectly valid terms in the domain of web discourse or more specifically in Twitter communication.

In order to see how detected out-of-vocabulary words were distributed among and within these 3 major groups, we manually analyzed all OOV tokens which appeared in text more than once and also looked at 1,000 randomly selected hapax legomena. The results of these estimations are shown and explained below.

We subdivided class 1 into the following subclasses:

1. regular German words, e.g. *aufm*, *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.

Though this division is admittedly arbitrary to a certain degree and also has the disadvantage of simultaneously involving different linguistic criteria, the underlying notion here was simple – 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).

In Table 1, percentage figures for each of the above subgroups are shown. 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 OOV-tokens (column "% of total OOVs") as well as with regard to their percentage rate in the list of only unique OOVs disregarding frequencies (column "% of unique OOVs").

Similarly to class 1, we subdivided the group "Sloppiness of users' input" into the following subgroups:

Table 1. Distribution of OOV words belonging to the class "Objective limitedness of MRD"

OOV subclass	`hunspell`		`TreeTagger`	
	% of total OOVs	% of unique OOVs	% of total OOVs	% of unique OOVs
regular German words	7.82	8.78	2.8	3.49
compounds	1.21	2.42	2.51	4.55
abbreviations	3.98	4.77	3.27	3.44
interjections	5.95	4.54	5.58	4.29
person names	4.73	6.41	2.32	3.47
geographic locations	1.5	2.53	1.16	1.88
company names	2.27	2.84	4.35	3.01
product names	2.13	2.57	2.45	3.23
newly coined terms	1.35	1.31	3.33	2.38
loanwords	3.68	4.03	3.29	2.87
foreign words	11.5	13.76	9.57	10.91
total	46.12	53.96	40.63	43.52

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, we explicitly marked each of these operations in our data. Statistical distribution of these subclasses is shown in Table 2.

Table 2. Distribution of OOV words belonging to the class "Sloppiness of users' input"

OOV subclass	`hunspell`		`TreeTagger`	
	% of total OOVs	% of unique OOVs	% of total OOVs	% of unique OOVs
insertions	0.49	1	0.18	0.34
deletions	8.44	6.38	6.52	5.27
substitutions	2.17	3.37	1.11	1.2
total	11.1	10.75	7.81	6.81

As is clear from the table, deletions are by far the most common type of misspellings. The reasons for that are either relatively frequent omissions of characters made by users or even more often automatic truncations of too long messages which are performed by Twitter service itself.[7]

[7] As is generally known, Twitter imposes a strong restriction on the length of posted messages which can be no longer than 140 characters. Upon exceeding this length, tweets get automatically truncated to the maximal allowed length.

Finally, the third group – "Stylistic specifics of text genre" – was subdivided into the subgroups:

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

according to the formal or lexical class which the tokens belonged to. Additionally, we also considered as *slang* spelling variants of standard language words which could be regarded as their colloquial equivalents. Such words included cases like, for example, *ned* instead of *nicht* or *grad* instead of *gerade* and were marked both as *misspelling*s and *slang* in our classification. A detailed statistics on the subgroups of class 3 is shown in Table 3:

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

OOV subclass	`hunspell`		`TreeTagger`	
	% of total OOVs	% of unique OOVs	% of total OOVs	% of unique OOVs
@-tokens	13.02	20.23	16.19	21.91
hashtags	7.35	6.18	13.06	10.59
links	2.43	0.4	4.89	6.07
smileys	2	0.73	6.88	1.2
slang	16.22	5.27	6.94	4.77
total	41.02	32.81	47.96	44.54

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 analyzed data, was recognized as out-of-vocabulary by `hunspell` but was not deemed as such by `TreeTagger`. Luckily, such singleton cases were rather rare exceptions during the analyses and did not affect much the remaining classes of OOVs, so that distributions of unknown tokens for both tools were more or less similar or at least comparable to a certain extent.

But an even more important conclusion which can be drawn from the analyzed data is the fact that for both `TreeTagger` and `hunspell`, Twitter-specific phenomena like special tokens, colloquial expressions or sloppily typed words accounted for more than a half of all unknown words found during the analysis. Since different classes of these Twitter phenomena were formed by different processes and also have different degrees of ambiguity, we will suggest different normalization procedures for each of them in the next Section.

4 Text Normalization Procedure

4.1 Replacement of Twitter-Specific Phenomena

In August 2007, Chris Messina, a renowned advocate of open-source community, sent out a Twitter message suggesting that the pound sign could be used for organizing groups on Twitter (cf. Parker, 2011). The "hash godfather", as Chris Messina described himself, allegedly borrowed this idea from Internet Relay Chats (IRC) – the forebears of modern social networks – which have been using the "#"-sign to mark channel names since the early 1990s.

Luckily for us, this "novelty" by Messina as well as @-tokens which are in Twitter to mark names of user account both have strict formal features by which these tokens could be identified. Other types of words which are usually considered as OOVs and have unambiguous formal traits are hyperlinks, e-mail addresses, and smileys. The presence of such formal criteria suggests that these classes could best be handled by rule-methods and namely finite-state techniques like finite-state transducers (FST) (cf. Jurafsky and Martin, 2008, pp.57-60).

For our purposes, we developed a prototypic Python system analogous to FSTs in which a set of regular expressions was associated with corresponding actions performed on matched subgroups. In this system, absolutely unambiguous tokens like web links and e-mail addresses were replaced with special words "%Mail" and "%Link" correspondingly. Emoticons were replaced with tokens "%PosSmiley", "%NegSmiley", depending on the type of emotion presumably conveyed by these expressions. In cases when an emoticon was ambiguous with regard to its polarity, it was replaced with the artificial word "%Smiley".

Additionally to that, problematic UTF-8 characters like special quotes and ellipsis which caused incorrect work of tokenizer were converted to their corresponding ISO-8859-1 equivalents.

For hashtags, we simply stripped off the pound character from the beginning of such words thus leaving only the alphabetic part of tokens. A more complicated case turned out to be the @-tokens. For them, we had to decide depending on the context whether these tokens played an important syntactic role in sentence or not. In cases they did, we additionally looked whether the alphabetic part following the "@" character was present in dictionary. If it was, we only stripped the "at" character off, otherwise whole word was replaced with token "%UserName". In cases when @-tokens served as addresses or retweet marks, we deleted them from message since they were of little interest and importance for further syntactic and semantic analysis and could hardly be assigned a proper part-of-speech tag or syntactic role.

All the artificially introduced words were added to custom user dictionaries of `TreeTagger` and `hunspell`. For tagging, we assigned NE tags to all artificial words except for smiley replacements which got the tag INJ. Additionally, positions and lengths of all replacements we remembered along with original words which were replaced and a restoration step to original forms was performed for certain classes of tokens after tagging.

4.2 Restoration of Umlauts

4.3 Squeezing of Elongated Words

4.4 Translation of Slang Idioms

5 Evaluation

6 Conclusion

Acknowledgement.

This work was financially supported by ... as part of collaborative project "...". The authors are also thankful to ... for help with the analysis 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

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

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

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)

Han, B., Baldwin, T.: Lexical Normalization of Short Text Messages: Makn Sens a #twitter. ACL HLT. (2011) 368–378

Jurafsky, D., Martin, J. H.: Speech and Language Processing. 2nd Edition. Prentice Hall (2008) 129, 323

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

Kukich, K.: Techniques for Automatically Correcting Words in Text. ACM Computing Surveys **24/4** (1992) 378–439

McVeigh, T.: Text messaging turns 20. The Observer (December 1, 2012)

Mukherjee, S., Malu, A., Balamurali, A. R., Bhattacharyya, P.: TwiSent: A Multistage System for Analyzing Sentiment in Twitter. In Proceedings of The 21st ACM Conference on Information and Knowledge Management CIKM 2012, Hawai, (Oct 29 - Nov 2, 2012)

McVeigh, T.: Text messaging turns 20. The Observer (December 1, 2012)

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

Parker, A.: Twitter's Secret Handshake. The New York Times, Page ST1, (June 10, 2011)

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)

Sparck Jones, K., Galliers, J. R.: Evaluating Natural Language Processing Systems. An Analysis and Review. Lecture Notes in Computer Science 1083, Springer. (1996)

Terdiman, D.: Report: Twitter hits half a billion tweets a day. `http://timmurphy.org/2009/07/22/line-spacing-in-latex-documents/`,Accessed February 26, 2013. (2012)

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