Rule-Based Normalization of German Twitter Messages

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. In this article, we conduct quantitative and qualitative analyses of unknown words in German Twitter messages, and propose a normalization method which prepares German Twitter text 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 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 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

The earlier works on normalization of noisy messages were greatly influenced by text restoration techniques commonly applied in the domains of speech and optical character recognition. One of the most well-established techniques in those fields at the beginning of the 1990-s was the method of "noisy channel model" (NCM) first introduced by Shannon in 1948. In this method, an input text T was regarded as a distorted version of some initial clean language signal S. And in order to find the original sequence, one had to solve the equation $arg\,max\, P(S|T)$ by computing an equivalent expression $arg\,max\, P(T|S)P(S)$. In this last form, the term – $P(T|S)$ – was usually referred to as *error model*, i.e. one which showed how likely it was that the given non-standard form T could have resulted from the assumed standard counterpart S. The latter part of the equation – $P(S)$ – was called the *language model* and accounted for the probability of form S to appear in standard text in general. It is however not specified by NCM, what should regarded as language model, what has to be considered as an error model and eventually how a set of standard language equivalents S should be derived for given non-standard form T.

In 1991, Mayes, Damerau et al. suggested an NCM approach in which they derived the set S by applying one of the four operations: insertion, deletion or substitution of a single character or transposition of two adjacent characters to string T and then leaving only variants which were present in vocabulary. Original word T was also included in the generated set. As language model, the authors used word bigrams. As error model, a uniform distribution was used which assigned equal probabilities to all generated variants and a constant probability α to the original word.

As noted by Kobus et al. (2008), NTN methods relying on either graphematic or phrasal segments usually reveal 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, 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 are the approaches sug-

gested by Beaufort et al. (2010) for French, and Oliva et al. (2013) for Spanish. Since the peculiarities of short messages are often language-specific, we 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. [1] 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`[2] and 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 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`.

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

1. **Objective limitedness of machine-readable dictionaries (MRD)**. Among this group, we counted words of basic vocabulary which did not get into applications' MRD either because they supposedly were rare or did not exist at the time when dictionaries were created. Another reason for the 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. **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;
3. **Misspellings**. In the scope of this group, we considered incorrect spellings of words encountered in text.

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

[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.

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. *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.

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 on Page 5, 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 OOVs (column "% of total OOVs") as well as with regard to their percentage rate in the list of only unique unknown tokens disregarding their frequencies (column "% of unique OOVs").

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

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. Additionally, spelling variants of standard language words which could also be regarded as their colloquial equivalents like, for example, *ned* instead of *nicht* or *grad* instead of *gerade*, were considered by us both as *misspelling*s and as *slang* in our classification. A detailed statistics on the subgroups of class 2 is shown in Table 2:

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

Table 2. 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 set, was recognized as out-of-vocabulary by `hunspell` but was not deemed as such by `TreeTagger`.

Finally, the last group – "Sloppiness of users' input" – was split into the following subclasses:

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 3.

Table 3. 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.[4]

But an even more important conclusion that can be drawn from the analyzed data is the fact that for both `TreeTagger` and `hunspell`, Twitter-specific phenomena like specially marked tokens, colloquial expressions or sloppily typed words accounted for more than a half of all OOVs found during the analysis. And since different classes of these Internet-based communication phenomena were formed by different processes and also showed different degrees of ambiguity, they should most probably be differently normalized depending on the characteristics they show.

[4] 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.

4 Text Normalization Procedure

4.1 Replacement of Twitter-Specific Phenomena

According to Parker (2011), hasthags were presumably introduced by Chris Messina, a renowned advocate of open-source community, in August 2007. The "hash godfather", as Messina calls himself, allegedly borrowed the idea of marking relevant topics in messages with the "#"- sign from Internet Relay Chats (IRC) – the forebears of modern social networks – which have been using the pound character for marking their channel names since the early 1990s.

Luckily for us, this markup "novelty" brought along a strict formal feature by which future hashtags could be identified. Other types of words which are also usually considered as OOVs in Twitter but have unambiguous traits in their written form are at-tokens, hyperlinks, e-mail addresses, and smileys. The presence of such formal criteria suggests that these classes could best be handled by rule-based 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 an FST in which a set of regular expressions was associated with corresponding actions performed on matched subgroups. In this system, unambiguous tokens like, for example, e-mail addresses were replaced with an artificial word "%Mail". Emoticons were substituted by tokens "%PosSmiley" or "%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 special word "%Smiley". Furthermore, leading "#" characters were stripped off from the beginning of hashtags thus leaving only the alphabetic part of these tokens.

A more complicated case turned out to be the at-mentions and hyperlinks. For them, we had to disambiguate whether these words played an important syntactic role in sentence or could be easily omitted without breaking the syntactic structure. In the former case, these words were replaced with artificial counterparts. In the latter case, such words were deleted from sentence in order not to confuse further intended parsing processing.

We measured on 1,000 sentences how well our system could disambiguate contexts for the two above-mentioned classes and also how well it could recognize emoticons and assign them correct polarity. Our estimated precision, recall and F1 metrics are shown in Table 4.

Table 4. Precision, recall and F1 score for replacement of Twitter-specific phenomena

OOV subclass	Precision	Recall	F1 score
at-mentions	98.87	98.87	98.87
links	99.44	100.00	99.72
smileys	93.79	76.26	84.12

Additionally, all the artificial words were added to custom user dictionaries of `TreeTagger` and `hunspell`. For tagging, we assigned NE tags to all artificial tokens with the only exception of emoticon replacements which got the tag INJ. Furthermore, positions and lengths of all replacements we remembered along with the original words which were replaced, and a restoration step to original forms was performed for user names after tagging.

4.2 Restoration of misspellings

A closer look at unknown words classified as misspellings revealed that a prevailing majority of them was considered as colloquial spelling variants. Such variants accounted for 70.5 % of detected spelling mistakes in `hunspell` data and for 66.17 % of misspellings recognized by `TreeTagger`. Frequency distributions of colloquial and non-colloquial misspellings in data analyzed with either tools are shown in Figures 1 and 2.

Fig. 1. Frequency distributions of non-standard spellings in data analyzed with `hunspell`

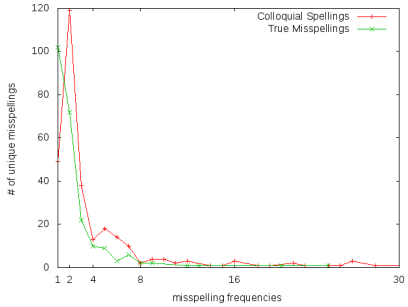
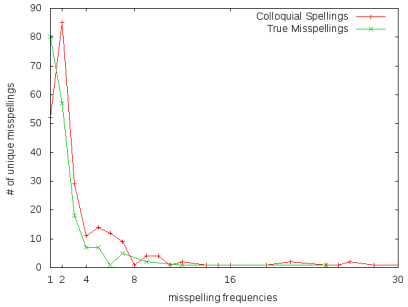


Fig. 2. Frequency distributions of non-standard spellings in data analyzed with `TreeTagger`



As can be seen from the figures, both kinds of spelling variations have nearly Zipfian distribution. However colloquial spelling variants, if used, tend to be used frequently whereas non-colloquial misspellings are rather represented by sporadic singletons. This could be explained by the fact that colloquial spellings are usually formed by systematic rewriting processes which are also normally applied to frequently used words. True misspellings, on the contrary, are rather produced by occasional slips of the finger, so they neither have an apparent system in them nor they tend to reoccur in text.

On closer examination of colloquial spellings, it could be found that a large part of them was a result of phonetically motivated rewritings.

According to our data, the most productive of such slang producing processes turned out to be:

- 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*;
- Complete omission or replacement of final consonants with their voiceless equivalents, e.g. *nich* instead of *nicht* or *Tach* instead of *Tag*;
- Omissions of 'ei' from indefinite articles, e.g. *ne* instead of *eine* or *nem* in lieu of *einem*;
- Multiple repetions of characters as a means of expressing elongation of sounds, e.g. *Hilfeeee*, *süüüß*;

Since all of the above processes followed some specific formation patterns. We developed a set of reverese transformation rules which first captured tokens with suspicious character sequences, checked them in the dictionary, and if these words were not found there, a transformation associated with each matched incorrect character sequence was applied.

TODO: more detailed and coherent description of rule module

Unfortunately, spelling variants which were classified as true misspellings did not show any regularities except for cases of incorrect spelling of sharp "s" and umlauts. While writting of "ss" instead of sharp "s" could also be considered as correct variant with regard to the Swiss norm. For umlauts, a straightforward replacement of character sequences like "ae", "oe" or "ue" because these character sequences also appeared in regular German words like "israelisch" or "quer". Therefore a set of exceptional cases had to be found before applying substitutions. As a possible source of such exceptions we considered words classified by us as in-vocabulary or valid out-of-vocabulary terms. Another source of possible exceptions was a corpus of newspaper texts which by definition were supposed to contain only proof-read content. To derive exceptional character sequences, we subsequently expanded the substrings to the left and to the right or simultaneously to both directions until these subsequence only captured words present in our "normal" data without capturing any misspeling from the list. After having collected exceptions, we only applied replacement operation to text spans lying in between exceptional substrings. On another test set of 500 cases, this method turned out to have made only one mistake. It replaced the word "zuende" with "zünde" in the sentence "RT @Netznarr: Als Amtsträger weißman, dass es zuende geht, wenn man in Umfragen unbeliebter als Westerwelle ist. #Wulff"

TODO: should be described as algorithm, should the same approach be tried for other misspelling char sequences?

5 Evaluation

To evaluate our system, we first performed an intrinsic evaluation by measuring the rate by which the amount of unknown tokens was reduced for `TreeTagger` and `hunspell`.

TODO: figures for OOV rates after normalization.

To comply with metrics commonly used for evaluating text normalization procedures for other languages, we also measured word error rate (WER) and sentence error rate (SER) for restoration of non-standard and incorrect spellings.

TODO: WER and SER figures after normalization.

Finally, as a first step towards an extrinsic evaluation, we measured how `TreeTagger` performance changed after normalization step was applied

TODO: Figures for tagging quality before and after normalization.

6 Conclusions and Future Work

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

We also 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 could effectively address some ofthe 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 at `http://dasom.ling.uni-potsdam.de/` under the terms consistent with Twitter regulations. Our classified data could, in our opinion, significantly save tedious manual work for other researchers.

Admittedly, our method still could be further refined and improved. As possible steps for such refinement, we see a possible addition of a machine-learning classifier which could help distinguish spelling mistakes from valid unknown words. On the other side, a better disambiguation of suggested replacement variants for misspellings could be achieved by letting 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 run. At any rate, these steps are beyond the scope of this work. But we hope to have laid the groundwork for them.

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

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

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)

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)

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

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