Normalisation and Analysis of Social Media Texts (NormSoMe)

Workshop Programme

28 May, 2016 (morning session)

9:00 – 9:05 – Introduction to the Workshop (by Andrius Utka)

9:05 – 9:45 – Session 1 (Chair: Martin Volk)
Torsten Zesch (keynote speech): *Your noise is my research question! - Limitations of normalizing social media data*

9:45 – 10:35 – Session 2 (Chair: Michi Amsler)
Judit Ács, József Halmi: *Hunaccent: Small Footprint Diacritic Restoration for Social Media*
Andrius Utka, Darius Amilevičius: *Normalisation of Lithuanian Social Media Texts: Towards Morphological Analysis of User-Generated Comments*

10:30 – 11:00 Coffee break

11:00 – 13:00 – Session 3 (Chair: Andrius Utka)
Jaka Čibej, Darja Fišer, Tomaž Erjavec: *Normalisation, Tokenisation and Sentence Segmentation of Slovene Tweets*
Hans van Halteren, Nelleke Oostdijk*: Listening to the Noise: Model Improvement on the Basis of Variation Patterns in Tweets*
Rob van der Goot: *Normalizing Social Media Texts by Combining Word Embeddings and Edit Distances in a Random Forest Regressor*
Ronja Laarmann-Quante, Stefanie Dipper: *An Annotation Scheme for the Comparison of Different Genres of Social Media with a Focus on Normalization*
Tatjana Scheffler, Elina Zarisheva: *Dialog Act Recognition for Twitter Conversations*

Editors

Andrius Utka	Vytautas Magnus University, Kaunas
Jurgita Vaičenonienė	Vytautas Magnus University, Kaunas
Rita Butkienė	Kaunas University of Technology

Organizing Committee

Andrius Utka	Vytautas Magnus University, Kaunas
Jolanta Kovalevskaitė	Vytautas Magnus University, Kaunas
Danguolė Kalinauskaitė	Vytautas Magnus University, Kaunas
Martin Volk	University of Zurich
Rita Butkienė	Kaunas University of Technology
Jurgita Vaičenonienė	Vytautas Magnus University, Kaunas

Workshop Programme Committee

Darius Amilevičius	Vytautas Magnus University, Kaunas
Michi Amsler	University of Zurich
Loic Boizou	Vytautas Magnus University, Kaunas
Gintarė Grigonytė	Stockholm University
Jurgita Kapočiūtė-Dzikienė	Vytautas Magnus University, Kaunas
Tomas Krilavičius	Vytautas Magnus University, Kaunas
Joakim Nivre	Uppsala University
Raivis Skadinš	Tilde, Riga, Latvia
Andrius Utka	Vytautas Magnus University, Kaunas
Martin Volk	University of Zurich

Table of contents

Workshop Programme…………..……………………………………………………………...............i
Programme Committee…………………… ………………………………………………...............ii
Table of Contents… ……………………………………………………………………...............iii
Author Index……………………………………………………………………………...............iv
Preface……………………………………………………………..…………………...............v
Judit Ács, József Halmi: *Hunaccent: Small Footprint Diacritic Restoration for Social Media*.............1
Jaka Čibej, Darja Fišer, Tomaž Erjavec: *Normalisation, Tokenisation and Sentence Segmentation of Slovene Tweets*……………………………………………………………………………..5
Rob van der Goot: *Normalizing Social Media Texts by Combining Word Embeddings and Edit Distances in a Random Forest Regressor*…………………………………………………………11
Hans van Halteren, Nelleke Oostdijk*: Listening to the Noise: Model Improvement on the Basis of Variation Patterns in Tweets*………………………………………………………………...15
Ronja Laarmann-Quante, Stefanie Dipper: *An Annotation Scheme for the Comparison of Different Genres of Social Media with a Focus on Normalization*……………………………………………23
Tatjana Scheffler, Elina Zarisheva: *Dialog Act Recognition for Twitter Conversations*...................31
Andrius Utka, Darius Amilevičius: *Normalisation of Lithuanian Social Media Texts: Towards Morphological Analysis of User-Generated Comments*………………………………………………..39

Author Index

Ács, Judit……1
Amilevičius, Darius……39
Čibej, Jaka……5
Dipper, Stefanie……23
Erjavec, Tomaž……5
Fišer, Darja……5
van der Goot, Rob……11
Halmi, József……1
van Halteren, Hans……15
Laarmann-Quante, Ronja……23
Oostdijk, Nelleke……15
Scheffler, Tatjana……31
Utka, Andrius……39
Zarisheva, Elina……31

Preface

Social media texts provide large quantities of interesting and useful data as well as new challenges for NLP. Social media texts include chats, online commentaries, reviews, blogs, emails, forums, and other genres. Typically, the texts are informal and notoriously noisy. Thus, many NLP tools have difficulties processing and normalizing the data.

As English social media has been investigated most widely, we also invited papers on other languages, especially those rich in inflections and diacritics, which cause additional processing problems. In the programme of NormSoMe, besides English, papers on Dutch, German, Hungarian, Lithuanian, and Slovene are included.

The workshop is aimed at researchers who have solutions, insights, and ideas for tackling the processing of social media texts, or who are interested in this field of research.

Hunaccent: Small Footprint Diacritic Restoration for Social Media

Judit Ács and József Halmi
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
`judit@aut.bme.hu, halmi.jozsef.ferenc@gmail.com`

Abstract

We present a language-independent method for automatic diacritic restoration. The method focuses on low computational resource usage, making it suitable for mobile devices. We train a decision tree classifier on character-based features without involving a dictionary. Since our features require at most a few characters of context, this approach can be applied to very short text segments such as tweets and text messages. We test the method on a Hungarian web corpus and on Hungarian Facebook comments. It achieves state-of-the-art results on web data and over 92% on Facebook comments. A C++ implementation for Hungarian diacritics is publicly available, support for other languages is under development.

Keywords: diacritic restoration, Hungarian, small devices, small footprint

1. Introduction

Diacritic restoration is the task of inserting missing diacritics in languages that have diacritically marked character in their orthography, but the diacritics are replaced with their corresponding Latinized grapheme for technical reasons, such as the lack of a specialized keyboard. Although human competence can easily restore diacritics real-time while reading, morphological, phonological and lexical information used by language technology is lost when accents are missing.

Most diacritic restoration methods are either dictionary-based (Kornai and Tóth, 1997) or grapheme-based (Mihalcea, 2002), (De Pauw et al., 2007). A decision list based approach was presented by (Yarowsky, 1999).

Recently, (Novák and Siklósi, 2015) presented an SMT-based approach for Hungarian combined with a morphological analyzer. They report up to 99.06% accuracy. (Zainkó and Németh, 2010) report 98% accuracy with a dictionary-based solution. Unfortunately, these systems are not available for download and components of the systems are non-free, therefore we could not reproduce them. To our knowledge, there are only two existing publicly available Hungarian diacritic restoration systems, one presented by (Kornai and Tóth, 1997), which is dictionary based, with a clever hashing solution to avoid excessive memory usage. Although the memory issues dealt with in the paper are no longer a concern, the agglutinative morphology of Hungarian still renders building a comprehensive word list very difficult. The other system is *charlifter* (Scannell, 2011).

2. Hungarian diacritics

Standard Hungarian uses 14 vowels, out of which 9 are diacritically marked in a symmetrical system (see Table 1). Five short vowels fit in the ASCII character set and two other short vowels do not. All long vowels fall outside ASCII. When Latinized, *á*, *é*, *í*, *ó* and *ú* are replaced by *a*, *e*, *i*, *o* and *u*, and *ő, ö* and *ű, ü* by *o* and *u* respectively. No consonants are diacritically marked. These vowels constitute 11.17% of Hungarian letters and more than 40% of Hungarian words contain at least one diacritic. In addition, Hungarian has two graphemes that are almost exclusive to Hungarian, *ő* and *ű*, and therefore the double acute accent is sometimes called *Hungarumlaut* by typographers.[1] The ISO 8859-2 and the Unicode character set support for *ő* and *ű* but *õ* and *û* are sometimes used as replacements for *ő* and *ű* or are mistakenly displayed since their codepoints in ISO 8859-1 correspond with the codepoints of *ő* and *ű* in ISO 8859-2. Other methods to avoid character-set confusion or deal with the lack of a non-ASCII keyboard are flying diacritics (*ő=o", ű=u"*) or telegram style (*ö=oe, ü=ue* etc.), but these conventions are less used nowadays and we do not address them in this paper.

Table 1: Hungarian vowels

short	a	e	i	o	ö	u	ü
long	á	é	í	ó	ő	ú	ű

Table 2: Diacritic statistics on 100M Hungarian words

non-whitespace tokens	94,365,073
types	2,230,835
accented ratio	40.77%
LexDif	1.017,9
ambiguous word type ratio	5.69%
non-ascii character ratio	11.333%

Table 2 illustrates vowel statistics computed on the first 100M (94M non-whitespace) tokens of the Hungarian Webcorpus (Halácsy et al., 2004; Zséder et al., 2012). More than 40% of tokens contain at least one accented vowel. *LexDif* (De Pauw et al., 2007) is the average number of orthographic alternatives per Latinized word. 5.69% of all word types have a non-unique inverse Latinized form. Table 3 lists the frequency and the Latinized form of each vowel.

3. Hunaccent

We present an ngram based approach without employing any kind of dictionary.

[1] *ő* is sometimes used in Faroese as well.

Table 3: Frequency of Hungarian diacritics and their Latinized form

Vowel	Latinized	Frequency
a		8.3616%
á	a	3.4328%
e		9.6705%
é	e	3.3895%
i		3.9559%
í	i	0.6142%
o		3.7476%
ó	o	0.9623%
ö	o	1.0095%
ő	o	0.8972%
u		0.9543%
ú	u	0.2615%
ü	u	0.5603%
ű	u	0.1894%

3.1. Data

We use the the Hungarian Webcorpus (Halácsy et al., 2004; Zséder et al., 2012). The corpus is POS tagged and we filter all tokens tagged as punctuation, but ignore the tags otherwise as we do not want to employ a POS tagger to the final system. Characters of the text are mapped to a small subset using the following preprocessing steps:

1. the text is lowercased,
2. punctuation is replaced with _ ,
3. digits are replaced with 0,
4. non-ASCII characters are replaced with *.

Reducing the number of different charcters to 29 avoids having an excessive amount of features. Accents are removed before feature extraction.

3.2. Features

We treat the diacritic restoration as 5 separate classification problems according to the 5 vowel groups (see Table 3). In each group, the vowels have the same Latinized form, ending up in three binary classification problems and two 4-way classification problems. We assume that an accented grapheme is always Latinized to the same ASCII character which is true for Hungarian, but might not apply to other languages.

Similarly to (Mihalcea, 2002), (Mihalcea and Nastase, 2002) and (De Pauw et al., 2007) our features are character ngrams in a sliding window approach. Word and sentence boundaries are converted to a single space and the sliding window treats the space as any other character. We experiment with three families of classifiers: decision tree, logistic regression, and SVM, all available in scikit-learn (Pedregosa et al., 2011). It turns out that decision trees considerably outperform logistic regression and SVM both in speed and in performance. The other advantage of decision trees is that the method is very good at identifying important features while keeping the decisions easy to interpret.

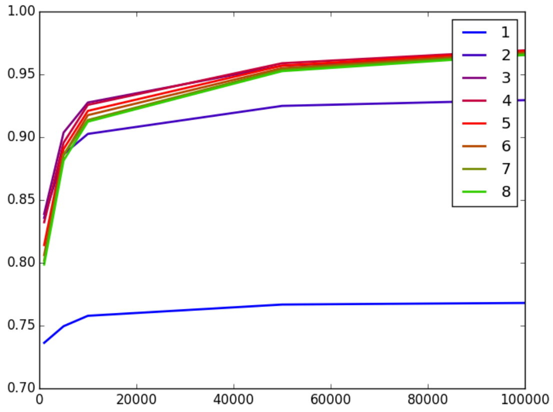


Figure 1: Average vowel accuracy with different window sizes and training data

We collect 2,000,000 occurrences of each of the 14 vowels and train on 90% and test on 10%. Hyperparameters include the sample-per-vowel count (varying from 1,000 to 2,000,000), the width of the sliding window, and the depth limit of the decision tree.

Table 4 shows the average accuracy of vowel classification taken over the 5 classification problems and weighted with the number of vowels in each class. We use symmetric sliding windows, meaning that a 4 window contains 4 characters before and 4 characters after the vowel. Accuracy improves with larger windows until a symmetric window of 4 is reached and consistently drops in each vowel class after that. These numbers are achieved without limiting the maximum depth of the decision tree. As 3 and 4 wide windows yielded the best results on 100,000 samples-per-class, we only trained with these windows on larger dataset. Memory limitation allows up to 2,000,000 samples-per-class, but in that case we had to limit the depth of the decision tree to 50.

Vowel-level accuracy is used for comparison and the best combinations are presented in Table 5.

We export the best scoring trees for each vowel group. These files are available in the *hunaccent* package and other languages will be added soon. In theory, an N deep deci-

Table 4: Vowel-level accuracy for different training sizes and window sizes

Window	Sample-per-vowel			
	5,000	10,000	50,000	100,000
1	74.95	75.78	76.68	76.8
2	88.64	90.26	92.49	92.95
3	**90.37**	**92.76**	95.69	96.55
4	89.56	92.57	**95.89**	**96.91**
5	89.05	92.1	95.7	96.85
6	88.57	91.75	95.5	96.74
7	88.62	91.37	95.38	96.63
8	88.14	91.23	95.26	96.54

Table 5: Word-level accuracy on Hungarian web corpora

Group	Depth	Sample size	Window	Acc
a	unlimited	1,000,000	4	98.92
e	50	2,000,000	4	98.17
i	50	2,000,000	4	99.63
o	unlimited	1,000,000	4	98.18
u	unlimited	500,000	4	98.61

sion tree has at most $2^{N+1} - 1$ decisions, but in practice, this number is usually much lower. The best configuration consist of 331,259 nodes for all groups, requiring a little bit over 5 MB RAM when loaded by the C++ implementation.

3.3. Accentizing social media

With a few exceptions (übra Adalı and Eryigit, 2014), diacritic restoration methods focus on well-formatted text such as newspapers or websites. As far as we know, this is the first attempt to perform it on Hungarian social media and this is why we prefer character-based methods. We used part of the Facebook comments collected by (Miháltz et al., 2015) for testing. The results are listed together with two Hungarian web corpora: the aforementioned WebCorpus and MNSZ2 (Oravecz et al., 2014)

3.4. Word-level accuracy

We compared 4 methods:

dictionary lookup retrieve the most common accentized form of every word. Leave OOV as it is. A 1,000,000 long frequency list is used.

ekito dictionary-based system by (Kornai and Tóth, 1997),

charlifter dictionary and character bigram-based system,

hunaccent our system.

Word-level accuracy was computed on a sample of 1,000,000 words on each dataset. Table 6 lists the results.

Table 6: Word-level accuracy on Hungarian web corpora

	Facebook	WebCorpus	MNSZ2
dictionary	92.67	96.98	**95.15**
ekito	92.61	94.45	93.2
charlifter	90.78	91.05	91.05
hunaccent	**92.77**	**98.36**	94.7

Table 7: Runtimes on 1M Facebook comments

Tool	Time
dictionary	26.5s
ekito	10s
charlifter	12s
hunaccent	**1.7s**

3.5. Limitations and drawbacks

The current method assumes a many-to-one mapping, where a single accented character is always mapped to single Latinized character, but more than one character may map to the same Latinized character. Tackling the issue of multicharacter mapping is out of the scope of this paper.

Another drawback of a character-based method is that non-existent word forms may be generated. Manual evaluation suggests that this is one of the largest error classes (see Section 4.).

The method does not recognize foreign words, which would be OOV in dictionary-based methods, and might accentize them (depending on the context, the English word *storage* is sometimes accentized as stóragé). A simple language model recognizing non-Hungarian words would probably help to solve this problem.

4. Manual evaluation

Manual evaluation was performed on accentized web corpora and Facebook comments using the dictionary-based and the grapheme-based methods. Considering that some words were incorrect in the input text, 4 outcomes are possible for each word: (i) correct input, correct output, (ii) incorrect input, correct output, (iii) correct input, incorrect output, (iv) incorrect input, incorrect output. Only those words were evaluated where the original and the output words differed.

The dictionary based method's errors were classified into the categories:

1. the input word is already incorrect, the output word is incorrect as well,
2. the input word is incorrect, but the diacritic restoration fixes it,
3. input word is out-of-vocabulary,
4. the input word's Latinized form is ambiguous and the wrong one is chosen from the dictionary.

Table 8 and table 9 illustrate the error classes on 1,000 manually annotated words.

Table 8: Error classes of the dictionary-based method on WebCorpus

Error class	Input	Output	Ratio
Incorrect input	írdogált	irdogalt	17.9%
Fixed input	Roviden	Röviden	8.8%
OOV	mérgesgázzal	mergesgazzal	40.5%
Ambigiuous input	feltéttel	féltettél	32.8%

Since the grapheme-based approach does not employ a dictionary, there are no OOV words, and non-existent word forms may be generated. As named entities constitute a considerable share of non-existent words, they were counted separately. Some Facebook users do not use accents, their comments were accentized and therefore differed in our output. This class is called *unaccentized input*. In some cases both the original and the output words were acceptable. Table 10 and Table 11 list the error classes on 200 samples from hunaccent's output.

Table 9: Error classes of the dictionary-based method on FB comments.

Error class	Input	Output	Ratio
Incorrect in	állapolgárok	allapolgarok	9.8%
Fixed input	boritékba	borítékba	14.75%
OOV	kormányváltók	kormanyvaltok	45.9%
Ambiguous	el	él	29.5%

Table 10: Error classes of hunaccent on MNSZ2

Error class	Input	Output	Ratio
Non-existent word	ez	éz	53%
Named entity	Theodorik	Theödorik	8%
Corrected	írdogált	irdogált	2%
Ambiguous input	igazat	igazát	36%
Incorrect input	ťhiábaŤ	ťhiábáŤ	1%

5. Conclusion and future work

We presented a small-footprint approach to diacritic restoration based on character ngrams features and using decision trees. Our experiments on Hungarian web corpora show that a symmetric 4 long sliding windows yield up to 98.36% word level accuracy and 98.88% character level accuracy when trained on a 2,000,000 sample-per-vowel dataset. We performed experiments on Hungarian Facebook comments and achieved 92.77% word-level accuracy even though the models were trained on web corpora and not social media.

Hunaccent outperforms dictionary-based approaches in all but one experiments and it is around a magnitude faster than every other tool with minimal memory footprint. The application of a moderate number of rules and a relatively short sliding window makes this approach well suited for mobile applications and social media where short texts are prevalent.

The system is available on GitHub.[2] and an Anrdoid client is under development.

6. References

De Pauw, G., Wagacha, P. W., and De Schryver, G.-M. (2007). Automatic diacritic restoration for resource-scarce languages. In *Text, Speech and Dialogue*, pages 170–179. Springer.

Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., and Trón, V. (2004). Creating open language resources for Hungarian. In *Proc. LREC2004*, pages 203–210.

Kornai, A. and Tóth, G. (1997). Gépi ékezés. *MAGYAR TUDOMÁNY*, 42(4):400–410.

Mihalcea, R. and Nastase, V. (2002). Letter level learning for language independent diacritics restoration. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7.

Table 11: Error classes of hunaccent on FB comments

Error class	Input	Output	Ratio
Non-existent word	ez	éz	39.5%
Named entity	Gyurcsány	Gyúrcsány	3.5%
Ambiguous word	számítana	számítaná	29%
Corrected	boritékot	borítékot	7.5%
Unaccentized input	sajat	saját	17%
Acceptable	hova	hová	3.5%

Mihalcea, R. F. (2002). Diacritics restoration: Learning from letters versus learning from words. In *Computational Linguistics and Intelligent Text Processing*, pages 339–348. Springer.

Mihiáltz, M., Váradi, T., Csertő, I., Fülöp, É., and Pólya, T. (2015). Beyond sentiment: Social psychological analysis of political facebook comments in hungary. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2015)*. ACL.

Novák, A. and Siklósi, B. (2015). Automatic diacritics restoration for Hungarian. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2286–2291.

Oravecz, Cs., Váradi, T., and Sass, B. (2014). The Hungarian Gigaword Corpus. In *Proceedings of LREC 2014*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Scannell, K. P. (2011). Statistical unicodification of African languages. *Language resources and evaluation*, 45(3):375–386.

übra Adalı, K. and Eryigit, G. (2014). Vowel and diacritic restoration for social media texts. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@ EACL*, pages 53–61.

Yarowsky, D. (1999). A comparison of corpus-based techniques for restoring accents in Spanish and French text. In *Natural language processing using very large corpora*, pages 99–120. Springer.

Zainkó, C. and Németh, G. (2010). Ékezetek gépi helyreállítása [automatic diacritic restoration]. In Géza Németh, G. O., editor, *A magyar beszéd: Beszédkutatás, beszédtechnológia, beszédinformációs rendszerek [The Hungarian Speech: Speech Research, Speech Technology]*.

Zséder, A., Recski, G., Varga, D., and Kornai, A. (2012). Rapid creation of large-scale corpora and frequency dictionaries. In *Proceedings to LREC 2012*, pages 1462–1465.

[2] https://github.com/juditacs/hunaccent

Normalisation, Tokenisation and Sentence Segmentation of Slovene Tweets

Jaka Čibej[1], Darja Fišer[1,2], Tomaž Erjavec[2]
[1] Dept. of Translation, Faculty of Arts, University of Ljubljana
Aškerčeva 2, 1000 Ljubljana
[2] Dept. of Knowledge Technologies, Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana
E-mail: jaka.cibej@ff.uni-lj.si, darja.fiser@ff.uni-lj.si, tomaz.erjavec@ijs.si

Abstract

Online user-generated content such as posts on social media, blogs, and forums, is becoming an increasingly important source of information, as shown by numerous rapidly growing NLP fields such as sentiment analysis and data mining. However, user-generated content is well-known to contain a significant degree of noise, e.g. abbreviations, missing spaces, as well as non-standard spelling, lexis, and use of punctuation. All this hinders the effectiveness of NLP tools when processing such data, and to overcome this obstacle, data normalisation is required. In this paper, we present a training set that will be used to improve the tokenisation, normalisation, and sentence segmentation of Slovene tweets. We describe some of the most Twitter-specific aspects of our annotation guidelines as well as the workflow of our annotation campaign, the goal of which was to create a manually annotated gold-standard dataset of 4,000 tweets extracted from the JANES corpus of Internet Slovene.

Keywords: normalisation, tokenisation, sentence segmentation, tweets, user-generated content

1. Introduction

With the rapid global expansion of the Internet, online user-generated content such as blogs, forums, and social media, is becoming an increasingly important source of information. The analysis of social media has become a popular research topic in a number of branches of NLP, including data mining, sentiment analysis, named entity recognition, and machine translation. However, user-generated content is well-known to contain a significant degree of noise, e.g. non-standard spelling and colloquialisms, frequent abbreviations, missing spaces and diacritics (Crystal, 2011; Eisenstein, 2013; Baldwin et al., 2013). In this regard, Slovene computer-mediated communication is no exception (Erjavec & Fišer, 2013; Zwitter Vitez & Fišer, 2015).

NLP tools trained on standard language data are less effective on noisy texts, which can be remedied through two different approaches: either by training new NLP tools on noisy data and adapting them to a particular variety of noisy language variety (see e.g. Yang & Eisenstein, 2013), or by improving the performance of existing NLP tools through data normalisation (Sproat, 2001). In the case of Slovene, a language with approximately 2 million speakers, developing new tools for its many regional and social language variants is unrealistic and unfeasible in terms of the available resources, so the logical step is to tackle noisy social media content via data normalisation.

In this paper, we present the compilation of a dataset that will be used to improve the tokenisation, normalisation and sentence segmentation of Slovene tweets in the context of the annotation of the JANES corpus of Internet Slovene (Fišer et al., 2015), a 160-million-token corpus of Slovene user-generated content containing tweets, forum posts, news site comments, and blogs.

The paper is structured as follows: in Section 2, we provide a brief overview of related work. In Section 3, we present the structure of the dataset to be annotated and the criteria used to compile it. We describe the annotation platform and the project workflow in Section 4 and then continue by describing the highlights of our annotation guidelines for sentence segmentation, tokenisation, and normalisation in Section 5. Finally, we conclude with a discussion of the results and suggestions for future work.

2. Related Work

Normalisation of Twitter content is not an uncommon task in the field of NLP. Approaches to the problem range from automatic construction of normalisation dictionaries to facilitate lexical normalisation through simple string substitution (Han et al., 2012); rule-based normalisation tackling omissions and repetitions in out-of-vocabulary tokens (Sidarenka et al., 2013; Clark & Araki, 2011); or normalisation using finite-state transducers (Porta & Sancho, 2013). In addition to normalisation models, language resources such as annotated datasets and corpora are also produced to help develop and test new normalisation systems (Alegria et al., 2014).

For Slovene, the most notable work so far for tweet normalisation is the normalisation model developed by Ljubešić et al. (2014), which aimed to improve the performance of existing Slovene text processing tools by training a character-level statistical machine translation system on a small manually validated lexicon containing pairs of original and normalised forms for the 1,000 most salient out-of-vocabulary (OOV) tokens with respect to a reference corpus. The model performed well, achieving a 69% accuracy when normalising OOV tokens, but there is still significant room for improvement. A major disadvantage of the system is that it is lexicon-based and does not take context into account when proposing normalisation. For this, an annotated corpus is required, the production of which is presented in this paper.

3. Dataset

A dataset of Slovene tweets to be manually annotated was prepared by extracting 4,000 tweets from the JANES corpus. The tweets were sampled according to their technical (T1–T3) and linguistic (L1–L3) standardness levels (Ljubešić et al., 2015), where 1 signifies a high degree of standardness and 3 a significant degree of non-standardness. For instance, a T1L3 tweet is standard from a technical perspective (punctuation, capitalisation, and use of spaces), but non-standard in linguistic terms (e.g. lexis and spelling), while a T3L1 tweet contains standard language written with e.g. no capital letters and no punctuation. A T3L3 tweet is non-standard in both regards.

- **T=1 / L=1**

Original: Bi kdo stanovanje v Kranju (Sejmišče) za 230€ na mesec? Starejše, enosobno (35m2), udobna kopalnica, visok strop, zastekljen balkon...
Standard: Bi kdo stanovanje v Kranju (Sejmišče) za 230 € na mesec? Starejše, enosobno (35 m2), udobna kopalnica, visok strop, zastekljen balkon ...
Characteristics
T: correct use of sentence-initial capitalisation and sentence-final punctuation, few missing spaces
L: completely standard lexis and spelling

- **T=3 / L=1**

Original: na sreco se motis,alkohol je v slo 100x vecji problem, primerjaj smrtnost in druzbeno skodo zaradi dovoljenih in nedovoljenih
Standard: Na srečo se motiš, alkohol je v Slo. 100x večji problem, primerjaj smrtnost in družbeno škodo zaradi dovoljenih in nedovoljenih.
Characteristics
T: no diacritics, no sentence-initial capitalisation, no sentence-final punctuation, missing spaces after punctuation
L: standard lexis and spelling

- **T=1 / L=3**

Original: Ja sej je blo to prav na koncu. Se mi je ena druga prijazna javla pa je rekla da sm prav poklical. Prvič ni šlo.
Standard: Ja saj je bilo to prav na koncu. Se mi je ena druga prijazna javila pa je rekla da sem prav poklical. Prvič ni šlo.
Characteristics
T: no missing spaces, correct use of punctuation and capitalisation
L: non-standard spelling (*sej* vs. *saj*, *blo* vs. *bilo*, *javla* vs. *javila*, *sm* vs. *sem*)

- **T=3 / L=3**

Original: jp,sis je se najbolj ziher... js sem se zarad tega 1x zastonj v portoroz peljala. mal na plazo pa tko.kaj pa 400 km :)
Standard: Jp, sis je še najbolj ziher ... Jaz sem se zaradi tega 1x zastonj v Portorož peljala. Malo na plažo pa tako. Kaj pa 400 km :)
Characteristics
T: no capitalisation (*portoroz* vs. *Portorož*), missing spaces before punctuation (*ziher... – jp,sis – tko.kaj*)
L: non-standard lexis (*ziher*, *jp*), non-standard spelling (*js* vs. *jaz*, *mal* vs. *malo*, *tko* vs. *tako*)

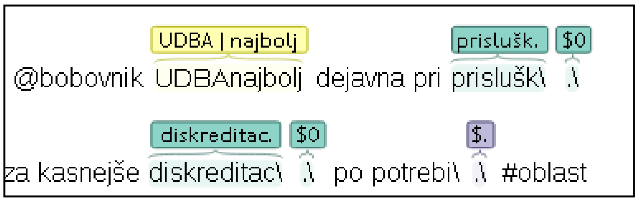
Figure 1. Examples of tweets with different standardness scores.

The dataset consists of four tweet categories, each contributing 1,000 tweets. The first three categories (T1L3, T3L1, and T3L3) contain tweets with the highest degree of non-standardness (either technical, linguistic, or both), while the last (T1L1) contains tweets that show next to no signs of non-standardness. Examples for each of these categories are shown in Figure 1.

The sampled tweets were automatically tokenised, segmented into sentences (Erjavec et al., 2005) and normalised (Ljubešić et al., 2014).

4. Annotation Platform

The dataset was divided into 400 files containing 10 tweets each and uploaded to WebAnno[1] (Eckart de Castilho et al., 2014), a general-purpose web-based annotation tool that enables multi-layer annotation. An example of annotations in WebAnno is shown in Figure 2. Yellow labels represent normalisation, green labels tokenisation, and purple labels sentence segmentation. We use special symbols to mark the deletion of a token ($0) and the end of the sentence ($.). A layer can also have multiple values (marked by "|") if a single input token should be split into more, or one word normalised into several.



@bobovnik UDBAnajbolj dejavna pri prisluškl .l
za kasnejše diskreditacl .l po potrebil .l #oblast

Figure 2: Annotations in WebAnno.

WebAnno was customised to allow for text annotations on the three layers relevant to our dataset: sentence segmentation, tokenisation, and normalisation.

If the same data is annotated by multiple annotators, the platform also offers a refereeing function, which enables a referee to compare multiple annotations in the same file and choose their final version.

The project workflow was designed to include a group of annotators and a referee with in-depth understanding of the annotation guidelines. The referee, who also managed the annotation campaign, designated a number of WebAnno files to each annotator group on a weekly basis. The end of each annotation phase was followed by a refereeing phase, during which the referee checked the annotations and, if necessary, provided constructive feedback to improve annotator performance by eliminating the most common and/or serious mistakes. If a particularly problematic issue arose during annotation, the annotation guidelines (see Section 5) were updated accordingly. The process was then repeated.

[1] https://webanno.github.io/webanno/

5. Annotation Guidelines

Based on a manual analysis of a small development set containing 200 randomly sampled tweets from all four categories in the dataset, annotation guidelines that address technical and linguistic aspects of the annotation process were prepared.

The technical guidelines covered the WebAnno annotation scheme and general aspects of working with the platform (joining or splitting tokens, deleting irrelevant and automatically generated tweets, dealing with complex multi-layer annotations, etc.), while the linguistic guidelines explained the criteria to follow when making language-related annotation decisions. The linguistic guidelines are summarized in the subsections below.

5.1 Sentence Segmentation

When determining sentence segmentation in tweets, the main criterion to consider is sentence-final punctuation (e.g. full stop, exclamation or question marks, two, three or multiple dots, quotes). However, tweets contain several other elements that may either appear next to sentence-final punctuation or, in its absence, fulfil a similar role. These elements are:

a) emoticons or emojis (;) =D ☺)
b) hashtags (#justsayin)
c) mentions (@author), and
d) URLs (http://t.co/fqVqV92mzc).

In the absence of sentence-final punctuation, these elements can effectively end a sentence. If the sentence ends with a series of elements, the final element is considered the end of the sentence[2]:

> Liverpool zasluženo owna Twitter, ampak na vrhu je pa fucking Iago Aspas hahaha :) #nogomet #LFC #SOULIV http://t.co/LCyEvyoVD7 ¶

If appearing after a sentence-final punctuation mark, these elements (or a series thereof) form an independent sentence:

> Življenje Je Cirkus. js sm pa čefur. Luka Stigl js sm se poscal v hlače k sm se vidu. bolano. ¶ :) ... ¶ http://t.co/QyzKRZqZnS ¶

5.2 Tokenisation

A number of elements were incorrectly split by the tokeniser and required corrections. These included abbreviations, emoticons, suffixes, and words including punctuation marks.

With abbreviations (*slov.* for *Slovene*), the tokeniser often interpreted the full stop as sentence-final punctuation and treated it as a separate token. In this case, the full stop needed to be joined with the abbreviation.

Emoticons often appeared in multiples with no spaces between and were commonly split by the tokeniser. Rather than treating each emoticon as an individual element, we decided to join the series into a single token:

- :) ¶ :) ¶ : ¶ * ¶ * → :):):**
- \ ¶ m ¶ / ¶ (¶ - ¶ _ ¶ - ¶) → \m/(-_-)

The same was done with suffixes as well as words that included punctuation:

- TV ¶ - ¶ ja → TV-ja
- sms ¶ - ¶ i → sms-i
- žen ¶ (¶ sk ¶) ¶ am → žen(sk)am
- politik ¶ (¶ e ¶ / ¶ o ¶) → politik(e/o)

5.3 Normalisation

With normalisation, two categories of words proved to be particularly problematic: non-standard words with multiple spelling variants, and foreign language elements.

5.3.1. Non-Standard Words with Multiple Spelling Variants

The first category includes non-standard words with no direct standard equivalent and multiple spelling variants (e.g. *orng*, *ornk*, *oreng*, *orenk* 'very' and *fovš*, *favš*, *fouš*, *fauš*, *fowš*, 'envious' or 'incorrect'). Such words are typically only used in spoken Slovene and have no standard spelling. In such cases, the JANES Tweet subcorpus was searched with regular expressions to find all possible spelling variants. The normalised form was then determined by selecting the most frequent one (in the above cases, *ornk* and *fouš*).

5.3.2. Foreign Language Elements

The second category consisted of foreign language elements with various degrees of adaptation to the Slovene language system in terms of spelling and morphology (e.g. *updateati*, *updajtati*, *updejtati*, *apdejtati*, 'to update'). Because of Slovene morphology, normalising these with their original language forms proved problematic (e.g. *poapdejtati*, *po-apdejt-ati*, 'to update') as it would involve introducing artificial forms absent in real language use (e.g. *po-update-ati*).

Because of this, foreign language elements were treated according to the following criteria:

a) if the word was spelled entirely phonetically (e.g. *dankešn*, 'danke schön', *aprišiejt* 'appreciate'), it would be treated as a Slovene non-standard word with multiple spelling variants (see section 5.3.1), and

b) if the word still exhibited any foreign language characteristics (e.g. non-Slovene letters or foreign language spelling), the normalised form would be the most frequent spelling variant in the JANES tweet subcorpus among those exhibiting foreign language characteristics (e.g. *updateati*, *updajtati*, *updejtati* → *updejtati*).

[2] In this paper, the end of a sentence or the delimitation between tokens is, where relevant, represented by the paragraph symbol (¶).

5.3.3. Exceptions to Normalisation

A number of Twitter- and CMC-specific elements such as mentions, hashtags, URLs, emoticons and emojis were exempt from normalisation and left in their original forms regardless of their (in)correctness.

In addition, normalisation did not extend to correcting syntactic mistakes (e.g. incorrect use of cases or mistakes in agreement, even if perceived as accidental), common lexical mistakes (e.g. using *moči* 'can' instead of *morati* 'must') or issues of style and register (*rabiti* 'to need (colloquial)' vs. *potrebovati* 'to need (standard)').

6. Annotation Campaign

In this section, we provide an overview and description of the phases of the annotation campaign.

6.1 Annotator Training

A two-day workshop was held in order to recruit annotators and familiarise them with WebAnno and the annotation guidelines. The workshop was attended by 11 annotators, all of them MA-level students of linguistics. The workshop consisted of a theoretical introduction to WebAnno, a hands-on tutorial, a presentation of the guidelines, and a training annotation session during which the participants annotated a small number of tweets. The goal of the annotation campaign was three-fold:

a) each tweet should be correctly segmented into sentences;
b) each tweet should be correctly split into tokens; and
c) all tokens should be normalised with the form closest to their standard equivalent (without radical changes to the word form, e.g. not substituting words with their standard synonyms); if the token is unclear or ambiguous, it should be left in its non-normalised form.

After the annotation session, a discussion was held to compare the annotators' decisions and the differences between them, as well as to provide correct solutions and the reasons for them in order to try and harmonise the annotators' decisions and raise inter-annotator agreement.

6.2 Annotator Testing

The workshop was followed by a test annotation session. The annotators were divided in two groups containing 5 and 6 annotators respectively. Each group was given 100 tweets from the test set and asked to correct the automatic annotations and add original annotations if necessary.

The annotations were then manually checked by the referee, who also evaluated the annotators' performance. Based on the evaluation results, 2 unreliable annotators were excluded from further assignments, and the guidelines were updated with several annotation issues that arose during the test session.

6.3 Annotation Phases and Annotator Performance

The annotation campaign was carried out in weekly phases from December 2015 to February 2016. The referee in charge of the campaign designated a number of WebAnno files to each group on a weekly basis. The remaining pool of annotators was divided into 3 groups consisting of 3 annotators.[3] A mailing list was created to allow annotators to ask questions and discuss problematic or borderline cases not included in the guidelines.

Annotator performance was monitored by measuring the annotators' effectiveness, i.e. the ratio between their annotation time and the number of tweets annotated (see Figure 3).

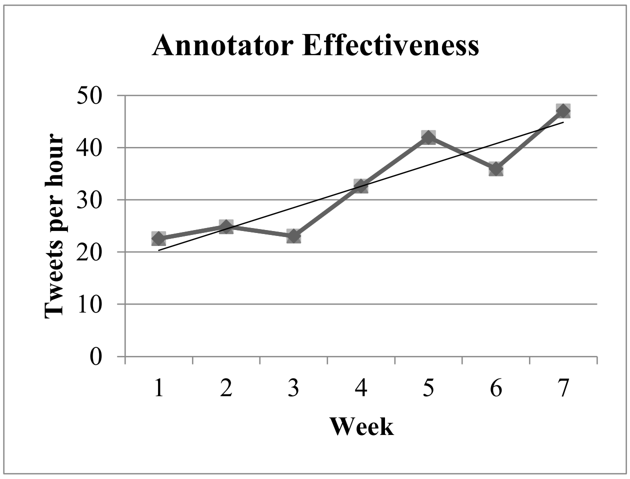


Figure 3: Annotator Effectiveness.

This was used to keep track of the annotators' weekly performance in order to optimize the flow of the annotation campaign. During the first three weeks, the annotators worked with non-standard tweets (T3L3) with a norm of 100 tweets per annotator per week. As the annotators grew more effective and the tweets steadily less noisy (T1L3, T3L1, and T1L1), the workload was increased to 150, 200, and finally 250 tweets per annotator per week. As can be deduced from Figure 2, in the case of Slovene, well-trained annotators can be expected to annotate approximately 35–45 non-standard tweets per hour – a significant improvement over the initial 21 tweets per hour.

The campaign took 7 weeks to finish, with a total of 272 hours invested by the annotators and 45 hours by the referee. On average, the annotators spent approximately 4.5 hours for each annotation session, and 30 hours for the entire campaign.

[3] When the annotators became more acquainted with the guidelines and three-member groups proved to be redundant, this number was reduced to 2, or, in the case of one accurate annotator, 1.

7. Results and Discussion

In Table 1, we give an overview of the amount of annotated tweets by standardness levels and overall. Of the initial sample of 4,000 tweets, 60 were discarded as irrelevant. In the rest, almost 10,000 sentences were identified, containing over 100,000 manually verified tokens or just under 86,600 words. It is noteworthy that the L3 tweets contain about 15% more words compared to the L1 ones. Overall, almost 12,000 words were normalised (14%), with T3L3 featuring a significantly higher number of normalisations (47%) than T1L1 (7%). The last two rows give the number of multiword normalisations, with either several original words being normalised to one word (e.g. *kvazi socializem* → *kvazisocializem*, *mega piksli* → *megapiksli*) or vice-versa (e.g. *nažalost* → *na žalost*, *nevem* → *ne vem*). The data shows that the latter category is far more frequent and also depends on the standardness level (unlike the first category).

	T1L1	T3L1	T1L3	T3L3	Total
Tweets	986	971	994	989	**3 ,940**
Sentences	2 ,413	2 ,009	2 ,934	2 ,620	**9 ,976**
Tokens	24 ,512	23 ,468	27 ,851	26 ,873	**102 ,704**
Words	20 ,333	20 ,190	22 ,912	23 ,159	**86 ,594**
Normalised words	887	1 ,136	4 ,251	5 ,570	**11 ,844**
Original multiwords	15	15	14	14	**58**
Normalised multiwords	27	63	109	139	**338**

Table 1: Quantitative Analysis of the Dataset.

During refereeing, a number of common sources of discrepancies between annotators arose. We provide a brief overview of the key problematic points for each layer in the following subsections.

7.1 Ambiguous Sentence Endings

In sentence segmentation, annotators were often faced with ambiguous sentence endings. The first category involves the use of two or multiple dots, as seen below:

> hah.. nvem ... to je pa čist odvisno od dneva ... hehe :)

The annotation guidelines required the annotators to interpret this ambiguous use of multiple dots either as a pause (which should be part of the sentence) or as sentence-final punctuation (which should end the sentence).

Similarly, in some cases, full stops, commonly used as sentence-final punctuation, were used in positions where a comma or space would be more appropriate, as seen below:

> @author1 . @author2 . @author3 . niti slucajno! kdo bo pa to placu?

A third category, especially in T3, included sentences that contained no sentence-final punctuation, but some other sign of sentence delimitation (e.g. a capital letter):

> Ko sm pa vidu to stran sm biu pa res vesel Čeprov ponavad nism za take fore :)

In many such cases, multiple (correct) interpretations were possible, which led to annotator disagreement. The final decision depended on the interpretation of the referee.

7.2 Words with Multiple Disambiguation Options

Annotators also faced ambiguity with normalisation. The most common example is the colloquial Slovene conjunction '*k*', which can be normalised to '*ko'* (when), '*ker'* (because), '*ki'* (which), and, more rarely, into '*kot'* (as) or '*kjer'* (where). The annotators were told to normalise '*k*' with the equivalent best suiting the context if possible, or to leave it in its non-normalised form if the interpretation was unclear.

A similar dilemma was posed by the word '*sm'*, which can be interpreted as either '*sem'* (I am), '*sem'* (here), or '*samo'* (only). Especially in short tweets, in which context was lacking, disambiguation proved difficult.

7.2 Misspelt Foreign Language Elements

Discrepancies between annotators were also frequent in the case of misspelt foreign language elements. According to the annotation guidelines, if a word exhibits characteristics of foreign language spelling, it should be normalised into the most frequent form exhibiting foreign language characteristics. If the word is completely foreign, it is normalised into its standard foreign language form. In the case of misspelt words like *lptop* (*laptop* vs. *leptop*) and *rter* (*router* vs. *ruter*), the annotators had to interpret the word either as foreign or as Slovene, most often by relying on the context.

7.3 Words of Ambiguous Origin

Several Slovene words, especially those containing the consonant cluster '*ks'* (*seks*, *indeks*) were often spelt using the foreign letter 'x' (*sex*, *index*). According to the annotation guidelines, Slovene words containing foreign letters should be normalised into the standard equivalents (e.g. *sex* → *seks*). Some annotators, however, interpreted these words as foreign words and left them unnormalised.

8. Conclusion

In this paper, we presented the dataset, annotation guidelines, and annotation campaign for the creation of a training dataset to be used normalisation, tokenisation, and sentence segmentation of Slovene tweets. In addition, we highlighted some of the more problematic annotation aspects which should be carefully considered when dealing with noisy social media text.

The next step in our annotation campaign will include expanding the annotated dataset with two other layers: morphosyntactic descriptions (fine grained PoS tags) and lemmas. We will also further extend the dataset to other social media text types, in particular forum posts and on-line comments.

The latest version of the annotation guidelines (in Slovene) is available at http://nl.ijs.si/janes/viri, and the annotated dataset will be made available via the CLARIN.SI language resource repository under the Creative Commons licence (CC BY-SA 4.0). The annotation guidelines have already been adapted for Croatian and Serbian, and similar annotation campaigns are currently on-going within the ReLDI project.[4] This will allow for a cross-lingual comparison of the datasets and their impact on tagging accuracy.

9. Acknowledgements

The authors would like to thank Špela Arhar Holdt, Kaja Dobrovoljc, Simon Krek, and Katja Zupan for their valuable contributions to the annotation guidelines, as well as the annotators who participated in this project: Teja Goli, Melanija Kožar, Vesna Koželj, Polona Logar, Klara Lubej, Barbara Omahen, Eneja Osrajnik, Predrag Petrović, Polona Polc, Aleksandra Rajković, and Iza Škrjanec.

The work described in this paper was funded by the Slovenian Research Agency within the national basic research project "Resources, Tools and Methods for the Research of Non-Standard Internet Slovene" (J6-6842, 2014–2017), and the Common Language Resources and Technology Infrastructure of Slovenia (CLARIN.SI).

10. References

Alegria, I., Aranberri, N., Comas, P. R., Fresno, V., Gamallo, P., Padró, L., San Vicente, I., Turmo, J., and Zubiaga, A. (2014). TweetNorm es Corpus: an Annotated Corpus for Spanish Microtext Normalization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC2014)*. ELRA, Reykjavik-Paris.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. (2013). How Noisy Social Media Text, How Diffrnt Social Media Sources. In *Sixth International Joint Conference on NLP*, pp. 356–364.

Clark, E., and Araki, K. (2011). Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. In *Procedia - Social and Behavioral Sciences 27*, pp. 2–11.

Crystal, D. (2011). *Internet Linguistics: A Student Guide*. New York: Routledge.

Eckart de Castilho, R., Biemann, C., Gurevych, I. and Yimam, S. M. (2014). WebAnno: a flexible, web-based annotation tool for CLARIN. In *Proceedings of the CLARIN Annual Conference (CAC) 2014*. Soesterberg, Netherlands.

Eisenstein, J. (2013). What to Do About Bad Language on the Internet. In *NAACL-HLT*. ACL, pp. 359–369.

Erjavec, T., and Fišer, D. (2013). Jezik slovenskih tvitov: korpusna raziskava. In *Družbena funkcijskost jezika: (vidiki, merila, opredelitve), Obdobja 32*. Ljubljana: Znanstvena založba Filozofske fakultete, pp. 109–116.

Erjavec, T., Ignat, C., Pouliquen, B., and Steinberger, R. (2005). Massive multi-lingual corpus compilation: Acquis Communautaire and totale. In *Proceedings of the 2nd Language & Technology Conference, April 21–23, 2005*. Poznan, Poland, pp. 32–36.

Fišer, D., Ljubešić, N., and Erjavec, T. (2015). The JANES corpus of Slovene user generated content: construction and annotation. In *International Research Days: Social Media and CMC Corpora for the eHumanities: Book of Abstracts, 23–24 October 2015*. Rennes, France, p. 11.

Han, B., Cook, P., and Baldwin, T. (2012). Automatically Constructing a Normalisation Dictionary for Microblogs. In *EMNLP-CoNLL 2012*. Jeju, Republic of Korea, pp. 421–432.

Ljubešić, N., Erjavec, T., and Fišer, D. (2014). Standardizing tweets with character-level machine translation. In *Computational linguistics and intelligent text processing: 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6–12, 2014: proceedings: part II, (Lecture notes in computer science, 8404)*. Heidelberg: Springer, 8404, pp. 164–175.

Ljubešić, N., Fišer, D., Erjavec, T., Čibej, J., Marko, D., Pollak, S. and Škrjanec, I. (2015). Predicting the level of text standardness in user-generated content. In *10th International Conference on Recent Advances in Natural Language Processing: Proceedings of RANLP 2015 Conference, 7–9 September 2015*. Hissar, Bulgaria, pp. 371–378.

Porta, J., and Sancho, J.-L. (2013). Word Normalization in Twitter Using Finite-state Transducers. In: *Tweet-Norm@SEPLN, Volume 1086 of CEUR Workshop Proceedings*. CEUR-WS.org, pp. 49–53.

Richard Sproat. 2001. Normalization of Non-Standard Words. In *Computer Speech & Language, 15(3)*, pp. 287–333.

Sidarenka, U., Scheffler, T., and Stede, M. (2013). Rule-Based Normalization of German Twitter Messages. In: *Proceedings of the GSCL Workshop Verarbeitung und Annotation von Sprachdaten aus Genres internetbasierter Kommunikation*, 2013.

Yang, Y., and Eisenstein, J. (2013). A Log-Linear Model for Unsupervised Text Normalization. In *EMNLP 2013*. ACL, pp. 61–72.

Zwitter Vitez, A., and Fišer, D. (2015). From mouth to keyboard: the place of non-canonical written and spoken structures in lexicography. In *Electronic lexicography in the 21st century*: *linking lexical data in the digital age*: *proceedings of eLex 2015 Conference*, *11-13 August 2015*, *Herstmonceux Castle*, *United Kingdom*. Ljubljana: Trojina, Institute for Applied Slovene Studies, Brighton: Lexical Computing, pp. 250–267.

[4] https://reldi.spur.uzh.ch/

Normalizing Social Media Texts by Combining Word Embeddings and Edit Distances in a Random Forest Regressor

Rob van der Goot
University of Groningen
R.van.der.Goot@rug.nl

Abstract

In this work, we adapt the traditional framework for spelling correction to the more novel task of normalization of social media content. To generate possible normalization candidates, we complement the traditional approach with a word embeddings model. To rank the candidates we will use a random forest regressor, combining the features from the generation with some N-gram features. The N-gram model contributes significantly to the model, because no other features account for short-distance relations between words. A random forest regressor fits this task very well, presumably because it can model the different types of corrections. Additionally we show that 500 annotated sentences should be enough training data to train this system reasonably well on a new domain. Our proposed system performs slightly worse compared to the state-of-the-art. The main advantage is the simplicity of the model, allowing for easy expansions.

Keywords: Normalization, Noise, Word Embeddings, Random Forest

1. Introduction

Because the task of normalization has a lot of similarities with the task of spelling correction, many of the same methods can be used. The standard framework for spelling correction consists of three steps: error detection, candidate generation and candidate ranking. In this paper, we will use this framework, but skip the step of error detection; because this model is meant to be used in a pipeline, this task can be postponed, so that a more informed decision can be made for this crucial step. Traditionally, the steps in this framework were based on a combination of lexical and phonetic distance measures. This approach was focused on spelling correction and motivated by the fact that every word that needs correction is a spelling error or typographical error. These alternations occur a lot in social media data, but are complemented by other types of alternations which are more domain specific. These include slang, abbreviations, domain-specific conventions and new linguistic structures.

In this work, we will expand a traditional spelling correction method to adapt to the noisy social media domain. Word embeddings are exploited to complement the traditional candidate generation which is based on lexical and phonetical distances. Furthermore, a random forest regressor will be used to combine the features which are mainly collected during the generation. This simple system allows for easy expansions, for example: multiword replacements, word deletion or word insertion. Evaluation will be done on the standard benchmark for normalization of English social media texts: LexNorm 1.2 (Yang and Eisenstein, 2013). An example sentence from this dataset is shown in Sentence 1.

(1) new pix comming tomoroe
new pictures coming tomorrow

2. Related Work

Han and Baldwin (2011) describe one of the first normalization approaches tailored for the social media domain. First, candidates are generated by finding lexically and phonetically close words. Ranking is then done with a support vector machine. A wide range of features is used: dependency tree distance, lexical edit distance, phonetic edit distance, prefix substring, suffix substring and the longest common substring.

A completely different approach is taken by Hassan and Menezes (2013). Here, a bipartite graph is used with on one side the words, and on the other side n-gram contexts in which these words occur. In this bipartite graph, Markov Random Walks are used to generate correction candidates. Ranking is done afterwards, based on a lexical similarity distance.

Xu et al. (2015) use lexical and phonetic features on the syllable level instead of the word or character level. Syllables are extracted from erroneous words and are converted to an ARPAbet representation (Rabiner and Juang, 1993). The ARPAbet encoding of the erroneous token can be compared to ARPAbet encodings of words taken from a dictionary. Edit distances on the ARPAbet encoding are then used to compare possible candidates.

An ensemble reranking method is proposed by Li and Liu (2014), where four different systems for normalization are combined including a spell checker and some machine translation methods. Building further on this work, Li and Liu (2015) created a joint model for normalization and POS tagging. The candidate lists of the reranking model discussed in the previous paragraph are used in a Viterbi decoding (Viterbi, 1973). Traditionally, all possible POS tags for a word in the sentence are used in the encoding, but in the new model all possible POS tags for all possible corrections are used in the encoding. This model achieves state-of-the art performance on the LexNorm dataset as well as on the standard benchmark for POS tagging of Twitter data (Owoputi et al., 2013).

3. Method

Our system is based on two steps: candidate generation and the ranking of the candidates. Both of them are discussed in more detail below.

3.1. Candidate Generation

Candidate generation for unintended disfluencies is a much studied problem; most approaches make use of the lexical or phonetic properties of a word to find similar words in a vocabulary. Due to the vast amount of work, and the good results on the task of finding lexically similar words, we consider this task to be as good as solved. For this reason, we will use the Aspell spell checker for this task, which achieves a recall of 98% on a list of common misspellings[1]. It uses a lexical edit distance combined with a phonetic edit distance based on the Double Metaphone algorithm (Philips, 2000). Aspell is slightly modified to be able to process words consisting of only one character and we include phonetic information about numerals.

To find normalizations replacements for intended noise, we need a more meaning-driven approach. Word embeddings capture the meaning of a word by using the context it occurs in. A big advantage of this method is the fact that word embeddings are trained on huge amounts of unlabeled data, which is readily available for the social media domain. Words that occur in similar contexts will be close to each other in the vector space, and are thus good normalization candidates. We will use the word embeddings model of Godin et al. (2015), which is originally used for named entity recognition for Tweets. This skip-gram model is trained on 400 million Tweets, uses 400 dimensions, and contains over 3 million types.

3.2. Candidate Ranking

The task of finding the correct candidate can be interpreted as a binary classification task as there are only two classes we are trying to distinguish: correct and incorrect. This interpretation enables the use of a binary classifier, but also introduces some problems. Firstly, a binary classifier can never guarantee to only assign one instance to a class, let alone a list of possible candidates. This is solved by ordering the candidates using the probabilities of being in the 'correct' class.

Secondly, the training of a classifier with very few instances in one class is a problem. Empirical experiments on our training data shows that 85% of all tokens should be left untouched, so simple ranking on one binary feature, and thus zeroing out the others, results in an accuracy above 85%. This is solved by removing the original word before the training of the classifier. Because the original word should often stay untouched, it is always used as the highest ranked candidate in the candidate list. Following from this, the ranking is only evaluated on the erroneous tokens, so only these tokens will be used as training data.

A Random Forest regression model is chosen because its structure can adjust well to the underlying problem. This model combines multiple decision trees, trained on random subsets of the training data. Each input will follow a path down in every decision tree resulting in a prediction value for each tree. These values are then averaged, which results in one final prediction. This model is very suitable, because the underlying problem is not binary; we are trying to normalize different types of disfluencies, which might have very different values for the different features. More concretely, this model can learn that a high value only on feature A can be enough to classify it as the correct candidate, without excluding that feature B can have the same effect. We use the Random Forest implementation of Scikit-Learn (Pedregosa et al., 2011) with its default parameters, except for the number of estimators, which is set to 100.

The following features are used to train the random forest model:

- A score used by Aspell to indicate the lexical and phonetical edit distance and binary features indicating if the candidate and the original word can be found in the Aspell dictionary.
- The distance in the vector space of the word embeddings model between the original word and the correction candidate.
- Uni- and bi- gram probabilities, taken from two different n-gram models: a noisy twitter n-gram model (Herdağdelen, 2013), and a model based on clean texts (Brants and Franz, 2006).

4. Evaluation

4.1. Data

Two different normalization datasets are used in this work:

- Train set: 2,577 Tweets annotated with normalization (Li and Liu, 2014). This dataset consists of tweets taken from the Edinburgh Twitter Corpus (Petrović et al., 2010), and are annotated using Amazon Turk.
- Test set: The LexNorm dataset (Han and Baldwin, 2011), 549 tweets from a different period annotated by different annotators.

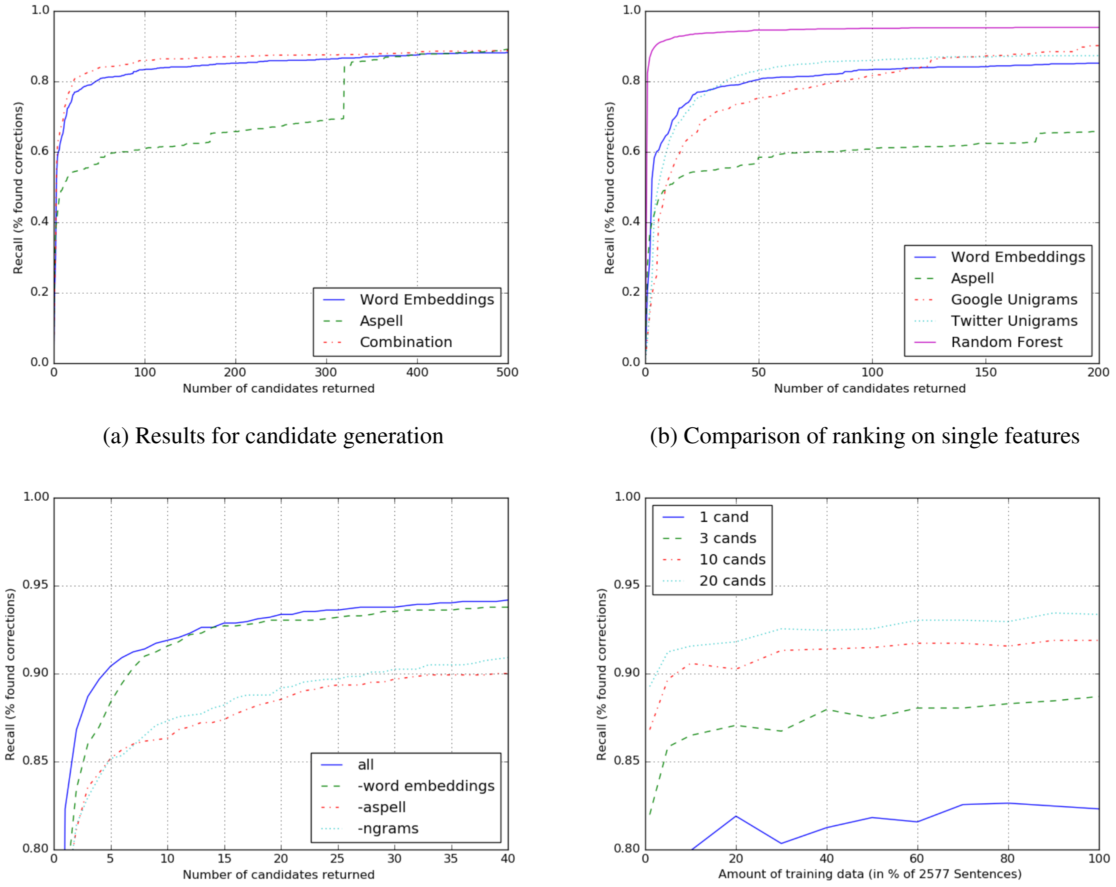
Both of these datasets use pre-tokenized tweets and only allow corrections on the word level. This setup ensures that our testing is robust with respect to biases in the annotation style and time period.

4.2. Generation

The generation is evaluated only on the words that are corrected in the annotated data. Two methods are compared, the traditional Aspell, and the generation from the word embeddings. However, our main interest is how well they can complement each other. For this reason, we included a naive combinatory method; this method simply takes equal numbers of candidates from the other 2 methods.

The individual and combined results are shown in Figure 1a. Word embeddings work better for this task than the traditional Aspell methods and combining them with a simple combination method already proves that they can complement each other. Additionally, we can see that the improvement in recall using a list of more than 100 candidates is moderate. One exception is the recall improvement for Aspell at 318 candidates. This is because at the Aspell candidate list for the common token 'u', the correct word

[1] http://aspell.net/test/orig/



(a) Results for candidate generation

(b) Comparison of ranking on single features

(c) Results of ablation experiments

(d) Effect of quantity of training data

Figure 1: Evaluation results of different experiments

'you' is at the 318th position. This word has already been found by the word embeddings (at position 3), so it does not affect the combination line. This one word accounts for a big part of the performance difference.

4.3. Ranking

Candidate ranking of a normalization system is usually only evaluated on the highest scoring candidate for each erroneous word. We will focus on two aspects: a high recall combined with a low number of candidates. A high recall ensures that the right candidate is in the list so that the following application has access to it and a low number of candidates is important for efficiency down the pipeline.

Table 1 shows a comparison of our system with the previous work that reports scores for different numbers of candidates, as well as the best performing system for this task. Note that our generation is slightly worse compared to the previous systems, this can probably be improved by adding some domain specific heuristics or by tweaking the word embeddings model. However, the ranking works surprisingly well, after only 20 candidates the upperbound is reached. The state-of-the-art system performs better on the top 1 candidate, but this system is a lot more complex. Unfortunately, the results for other numbers of candidates are not reported.

4.4. Feature Importance

First, we will evaluate how our features can perform on their own, then we will see how important the feature are with respect to the model in an ablation experiment.

Figure 1b shows the results of ranking on single features. Word embeddings have the best performance for a low number of candidates while the Twitter unigrams have the highest performance with more candidates. Additionally, we can see that Twitter N-grams generally work better, even though the google N-gram model is based on clean data. Presumably, this is because the Twitter N-grams have less sparsity with respect to the test data.

The ablation experiments are done on feature-groups. Grouping is done based on source level. The same grouping as in Section 3.2. is used. The results of the ablation experiments are shown in Figure 1c. Surprisingly, the word embeddings are the least important for ranking. This is

System	top1	top3	top10	top20	upper bound
Li and Liu (2012)	73.0	81.9	86.7	89.2	94.2
Li and Liu (2014)	77.14	86.96	93.04	94.82	95.90
Li and Liu (2015)	87.58				
Our system	82.31	88.70	91.89	93.37	93.37

Table 1: Recall of our system compared to previous work

probably because they reflect information from only one perspective, the distance in the word vectors, whereas the N-grams reflect on unigrams and bigrams from two different language models. Furthermore, the N-grams reflect on the relations of close words, which are important for grammatical correctness. Aspell appears to be very important for the ranking step, presumably because most types of alternations use some sort of lexical or phonetic variation of the intended word.

4.5. Reduce Training Data

To adapt this model to another domain, three resources are needed. A word embeddings model, an N-gram model and annotated data. Because an annotated dataset is the most expensive resource to acquire, only this resource is tested for quantity. Figure 1d shows how the performance drops when we decrease the amount of training data. After using 20% ($\approx$ 500 sentences) of the training data the improvements in performance are quite small.

5. Conclusion

We have shown that a spelling correction system can be converted to a normalization system by using modern techniques. Word embeddings can complement the lexical and phonetical approaches well for candidate generation because it targets other types of noise. Additionally, a random forest regressor can fit well to the normalization task, presumably because it can model the different kinds of noise in different parts of its trees. There are still plenty of improvements possible for this system, Aspell is not designed for this domain and the word embeddings model was preprocessed for another task. Almost no parameters have been tuned for the random forest regressor. Another source of improvement could be the addition of features.

Further directions include the addition of multiword replacements, but mainly the use of this system in a pipeline. Only then its usefulness can be properly tested. Our system outputs the whole candidate lists, and is made available on the authors website.

Acknowledgements I would like to thank Gertjan van Noord, my other colleagues and the anonymous reviewers for their valuable feedback, and the Nuance Foundation for funding the Parsing Algorithms for Uncertain Input project.

6. Bibliographical References

Brants, T. and Franz, A. (2006). Web 1T 5-gram version 1. Technical report, Google.

Godin, F., Vandersmissen, B., De Neve, W., and Van de Walle, R. (2015). Multimedia Lab @ ACL WNUT NER shared task: Named entity recognition for Twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China, July. Association for Computational Linguistics.

Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA, June. Association for Computational Linguistics.

Hassan, H. and Menezes, A. (2013). Social text normalization using contextual graph random walks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1577–1586, Sofia, Bulgaria, August. Association for Computational Linguistics.

Herdağdelen, A. (2013). Twitter n-gram corpus with demographic metadata. *Language resources and evaluation*, 47(4):1127–1147.

Li, C. and Liu, Y. (2012). Improving text normalization using character-blocks based models and system combination. In *Proceedings of COLING 2012*, pages 1587–1602, Mumbai, India, December.

Li, C. and Liu, Y. (2014). Improving text normalization via unsupervised model and discriminative reranking. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 86–93, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Li, C. and Liu, Y. (2015). Joint pos tagging and text normalization for informal text. In *Proceedings of IJCAI*.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia, June. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). The Edinburgh Twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26, Los Angeles, California, USA, June. Association for Computational Linguistics.

Philips, L. (2000). The double metaphone search algorithm. In *C/C++ users journal*, volume 18, pages 38–43.

Rabiner, L. and Juang, B.-H. (1993). Fundamentals of speech recognition. Technical report, Prentice hall.

Viterbi, A. (1973). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, 13(2):260–269.

Xu, K., Xia, Y., and Lee, C.-H. (2015). Tweet normalization with syllables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 920–928, Beijing, China, July. Association for Computational Linguistics.

Yang, Y. and Eisenstein, J. (2013). A log-linear model for unsupervised text normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 61–72, Seattle, Washington, USA, October. Association for Computational Linguistics.

Listening to the Noise: Model Improvement on the Basis of Variation Patterns in Tweets

Hans van Halteren, Nelleke Oostdijk
Radboud University Nijmegen, Dept. of Linguistics / CLST
P.O. Box 9103, NL-6500HD Nijmegen, The Netherlands
E-mail: hvh@let.ru.nl, N.Oostdijk@let.ru.nl

Abstract

In this paper, we take the view that the wide diversity in the language (use) found on Twitter can be explained by the fact that language use varies between users and from one use situation to another: what users are tweeting about and to what audience will influence the choices users make. We propose to model the language use of Twitter tribes, i.e. peer groups of users tweeting in different use situations. We argue that the use of tribal models can improve the modeling of the substantial variation present in Twitter (and other social media), and that the resulting models can be used in the normalization of text for NLP tasks. In our discussion of variation at the linguistic levels of orthography, spelling, and syntax, we give numerous examples of various types of variation, and indicate how tribal models could help process text in which such variation occurs. All examples are derived from our own experience with the Dutch part of Twitter, for which we could draw on a multi-billion word dataset.

Keywords: Twitter, social media, language variation, metadata induction, language modeling, spelling, syntax

1. Introduction

In many fields, data from the social media are judged to have enormous potential for research. At the same time, social media data are generally quite different from data originating from the traditional media. In many of the contexts of social media communication, the authors/users do not appear to feel bound to adhere to the norms that have been set for the standard language and deviate from these norms in their use of orthography, spelling, and/or syntax. Most of these deviations are intentional. In fact, they tend to follow conventions upheld within the authors' peer groups. This means that if we manage to identify the peer groups in question, we are able to model the variation to a large degree. This in turn leads to (a) better recognition of the factual information being transmitted as well as (b) information about the authors and their communicative goals as encoded in the variation.

In this position paper, we look at a specific type of social media data, namely text on the Dutch part of Twitter.[1] Now, in tweets we find a number of special communicative devices, either unique to Twitter or shared with other social media. Emoticons can be used to represent emotional content effectively, discussed topics can be marked with hashtags, other authors can be addressed or mentioned by quoting their username preceded by an at sign, and URLs can link to tweet-external additional content. How these devices are used by various groups is also an interesting subject of study. However, in this paper, we will ignore these devices and focus exclusively on linguistic objects already known in traditional text types.

In the following sections, we first explain our viewpoint in more detail (Section 2), after which we zoom in on separate linguistic levels, viz. orthography (Section 3), spelling (Section 4), and syntax (Section 5). Finally, we return to the overall picture for conclusions and a vision of the future (Section 6).

2. Twitter tribes

In recent years, numerous studies have been directed at the mining of social media data for various purposes. In most of these studies, the observation is made that texts in the social media are quite unlike texts published in traditional media and it is not uncommon to find that texts are being characterized as "noisy" and/or "to be corrected" (e.g. Kaufman and Kalita, 2010; Han and Baldwin, 2011). These days there is a lively research area both investigating the extent of the problem (e.g. Baldwin et al., 2013; Baldwin and Li, 2015) and committed to the attempt to extract as much information as possible despite the level of noise, using various methods as we can see e.g. in the report on a 2015 shared task on text normalisation for Twitter (Baldwin et al., 2015). The activity of the field is witnessed by the presence of a multitude of workshops, such as W-NUT, SocialNLP, NLPIT, and NormSoMe.[2]

While our work clearly falls in this research area, and also concerns the improvement of mining of social media text, our primary mining activities are linguistic studies, such as those into the linguistic variation in the social media (van Halteren & Oostdijk, 2015). Our perspective leads us to approach the task from a direction which is rather different from what appears to be mainstream, but which is closer to what we find in approaches adopted for example in Bryden et al. (2013)[3]

[1] That is, tweets by users communicating primarily in the Dutch language, most of whom are of Dutch or Flemish origin. The TwiNL dataset on which we draw is already collected in such a way that only a few percent of non-Dutch tweets remain (Tjong Kim Sang and van den Bosch, 2013), and we have managed to reduce that percentage to well under 0.5% (van Halteren, 2015).

[2] It is outside the scope of this paper to give a full inventory of such work here, and we limit ourselves to some examples. A good starting point for a deeper literature study would be the proceedings of the mentioned workshops.

[3] In fact, we adopted the term *Twitter tribes* as suggested by Jason Rodrigues in a Guardian blog about Bryden et al.'s work (http://www.theguardian.com/news/datablog/2013/mar/15/twitter-users-tribes-language-analysis-tweets).

and Eisenstein et al. (2014).

So far, in our research we have focused on Twitter as here data are available to us in large quantities (Tjong Kim Sang and van den Bosch, 2013). For any given research project, we first need to create a balanced corpus with reliable metadata. As all tweets carry a time stamp and many tweets are tagged for geolocation, efforts are mostly directed towards collecting additional metadata, viz. pertaining to the author and the use situation (topic, purposive role). As it turns out, author characteristics (gender, age) can to a fair degree be induced from the authors' language use (van Halteren & Speerstra, 2014; van Halteren, 2015). The same is true for the use situation. A special case here is the use of hashtags as a kind of user markup to indicate the main topic(s) of a tweet explicitly. However, this use of hashtags is mainly related to a specific type of Twitter discussion and is far less used in more personal tweets, i.e. the majority of tweets. This means that for topics too, we have to look at the contents, here topic-related words, rather than to the explicit metadata.

Now, on the one hand, for proper modeling of language variation and, on the other, for facilitating all mining tasks, we too want to identify some kind of "normal form" for social media text, both to be able to generalize away from individual forms and to be able to use available NLP tools. However, we feel that deriving such a normal form from the observed form can be informed by knowledge about the individual author, the peer group addressed, and the communicative goal. In the context of Twitter, we defined a "Twitter tribe" to be "a set of authors who share specific characteristics, discussing a set of related topics, in specific use situations"[4]. Such Twitter tribes can range from very narrowly focused, e.g. a specific community of twelve students discussing public transport, to very widely focused, e.g. all youngsters discussing any kind of topic. At both these levels of focus, we have found that there are measurable differences between tribes.[5] [6] As for the use situations, our investigations have so far been limited, especially since the use of Twitter is in itself already rather restricting the range of situations. However, we did observe that the language use in tweets carrying a hashtag, i.e. tweets aimed at a larger audience, conforms more to the standard language (van Halteren & Oostdijk, 2014).

Having only just proved the validity of the Twitter tribe concept, we did not yet implement a full tribal recognition engine, nor did we apply tribal models to text normalization. This means that as yet we cannot measure the potential quality improvement for any given task. However, we can give an overview of the types of variation we observed in our investigations,[7] and sketch how tribe modeling could be used to harness this variation.

3. Orthography

In traditional text types, we are used to a very specific markup system, with spacing separating words, punctuation indicating larger structural units, and capitalization fulfilling both lexical and structural functions. For most professional Twitter feeds, as well as many discussions by older users, we see that this markup is generally used in the standard fashion.[8]

Elsewhere, orthography appears to be far more random. Capitalization is generally ignored, or at most used to stress words. One reason may be that text input is not done with a standard keyboard with a simultaneous upper case key, but with some touch screen input method which toggles between separate upper and lower case keyboards. A similar situation exists for punctuation. Given the additional effort, and the fact that not using this standard markup does not seem to affect the interpretability of the message, many authors apparently decide just not to use the standard, as a side effect freeing capitalization for expressing stress. The effect when examining random tweet samples is that the use of capitals and punctuation appears almost random. Ideally, we should construct a tribe model, preferably modulated by a usage model for each individual character input method.[9] This, however, is still future work. For now, we are limited to trying to recognize that a specific user does not adhere to the traditional standards or conventions, and then (for this user) just assume that this component of the information in the message is not available.

For spacing, the input method does not seem to be the problem, as the space bar is almost always available.[10] Still, spacing as well is often different from

[4] We already have indications that the language use also changes over time. However, we decided not to include this factor in the definition of the tribe. We intend to study differences over time as a separate dimension, and view it as the evolution of each tribal language.

[5] As for narrowly focused tribes, we have investigated communities of authors (4-50 members) being in frequent contact, discussing the topic areas of school work, public transport, football, politics, and personal grooming (i.e. care for one's appearance, not to be confused with grooming in the internet predator sense). When comparing n-gram counts in which topic dependent words have been masked, there are (on average) significant differences in language use, both between discussions of different topics within each community, and between discussions of the same topic within different communities (van Halteren & Oostdijk, Submitted).

[6] As for widely focused tribes, we have shown differences in language use between the young and the old (van Halteren, 2015), as well as between men and women (van Halteren & Speerstra, 2014).

[7] Given that each investigation was quite extensive, we can only provide a summary in this paper, and will have to restrict ourselves to directing the reader to the individual publications for more details.

[8] With some exceptions, such as information feeds like job agencies and dating bureaus, which employ a more field-like structure in which spacing is sometimes omitted.

[9] Which input method is used can most often be deduced from the metadata in the Twitter JSON.

[10] The exception here might be voice input, and input method errors there should lead to more than just spacing problems.

the norm. We see both blanks left out and added where they are not needed. We investigated the extent of this phenomenon by manually annotating 1,000 tweets, randomly sampled from a year of tweets. In these tweets, we found some 300 cases of variant spacing in about 200 tweets. In most cases (60% of all variants), the variation was adjacent to punctuation. For processing, additional blanks in such contexts (36%) are completely unproblematic. Leaving out blanks where they are expected next to punctuation may sometimes lead to (mild forms of) ambiguity, e.g. where emoticons flow together with normal punctuation or where a word-period-word sequence might be mistaken for a URL, but generally this does not cause serious problems for processing.

More interesting are those cases where only words are adjacent to the variant spacing. In most cases where two or more words are merged (be it just glued together or fused more extensively), we found this is done deliberately (24%), possibly as a shortening mechanism. This is supported by the fact that there is quite some regularity here. We see that blanks are deemed superfluous within common bigrams, and that in many of these cases we see the formation of clitics (3%). In later investigations, we observed that even though cliticization occurs frequently the authors do seem to avoid ambiguity. As an example, *dat is* ('that is') can be shortened to *das*, and *dat ik* ('that I') to *dak*. Both of these shortened words are in the lexicon as an existing noun. *das* is both "badger" and "scarf" or "tie"; *dak* is "roof". Now, the alternative interpretations of *das* are needed much less frequently (in the case of scarf also because of more often used alternatives) than those of *dak*, and this difference is reflected in the usage of the shortened forms: if we examine the forms which are closest in terms of context vectors (using a window of two tokens left and two tokens right; cf. van Halteren, In prep.), *das* gives us a top-5 with *da's*, *dat's*, *dats*, *datis*, and *dass*, proving active use of the clitic, but *dak* gives us the top-5 *dakkie* (vernacular diminutive of *dak*), *balkon* ('balcony'), *dakje* (official diminutive of *dak*), *plafond* ('ceiling'), and *aanrecht* ('sink'), showing the clitic here is apparently shunned.

Such deliberate spacing variations can be lexicalized in the language use of specific tribes, leading to a situation much like that for spelling (Section 4). As an example, in one user community, we observed that the combination *maar ja* (lit. 'but yes', i.e. 'but well') was practically always written without a space; interestingly, the initial form *maarja* was over time more and more replaced by the even shorter *mja*.

In other cases of spacing variation between words (11%), we assume that the user is ignorant of the norm for spacing, e.g. when components of separable verbs are adjacent, or with compounds (which in Dutch should be written as a single word). Other categories of words where variant spacing is found include names, archaic forms, and words containing prefixes. In ignorance-related cases, the variation is typical for the user, but it sometimes propagates through conversations.

Finally there are cases (2%) where we did not identify any (apparent) regular system underlying variant spacing, and which might therefore just be typos.

Even though the majority of spacing variants appear to be resolvable, we think that here lies the hardest problem for proper processing, especially if one intends to use the traditional NLP architecture where tokenization is addressed in a separate preprocessing step.

4. Spelling

Regarding the spelling used by Twitter users, a random selection of tweets also tends to give the impression of almost random noise. However, if we investigate the data more extensively, and apply some classification, we start seeing patterns.[11]

As with orthography, there are large numbers of tweets, produced in a professional context or in the context of serious discussions between adults, where spelling usually conforms to the accepted norms for written language. Virtually all spelling deviations here are caused by typos; only in very few cases users appear to opt for a form of creative spelling. In some contexts, we do see extensive use of foreign words, but these too tend to follow standard spelling and topic-specific lexicons could be created. Alternative spellings are mostly found with younger and/or less educated users. But here too, we have the impression that each group of users mostly uses its own lexical and morphological conventions, picking mechanisms from the repertoire we describe below. Once we have determined what tribe we are dealing with, we can select the corresponding lexicons and rules for processing.

As already mentioned, there appears to be a fixed repertoire of mechanisms to vary spelling. However, before the discussion of this repertoire, we will first exemplify the level of variation with the word *school* ('school'), which we investigated when working on various techniques for modeling spelling variation. Table 1 shows the most frequent forms derived for school with a word form clustering algorithm using form relation information based on both contextual similarity and edit distance calculated with the Viterstein algorithm (van Halteren & Oostdijk, 2012). Figure 1 shows the forms that were only suggested for a single text instance to be connected to the same cluster. Apart from the forms shown, there were many more, leading to a total cluster of 507 forms. It should be noted that these 507 do appear to contain some false positives. In Table 1, we see the plural form *scholen*, as well as some other nouns with similar spelling, such as *schoot* ('lap').[12]

[11] For more quantitative information, and a description and evaluation of an early technique for spelling normalization for Dutch tweets, see van Halteren & Oostdijk (2012).

[12] Although *schol* is also a kind of fish ('plaice'), we do not think this should be counted as a false positive, given the distribution of discussion topics on Twitter.

8585 schooll	1643 sgoool	740 sgl	383 schoooooool	187 schooooooool
6245 schooool	1637 schoooooool	637 schoel	340 chool	179 schoooll
5468 sgol	1403 sschool	627 sgooll	277 schoop	169 sqool
5412 schol	1119 sxhool	549 scholl	276 skoool	161 scho
4926 shool	1011 schoolll	542 svhool	269 dchool	161 schoiol
3964 schoool	988 achool	529 schoot	260 schoolo	160 schoolx
3955 schoooool	981 scool	514 shcool	245 schooolll	159 schoola
3644 schhool	964 scholen	500 schoorl	231 schoor	156 sgoowl
2451 schhol	891 scchool	448 schoowl	220 schoolt	150 schiol
2410 schoooll	866 sgool	437 scgool	219 schoof	147 schhoool
2345 schook	768 schoolk	393 skool	214 schoolie	145 schiool
2323 schoo	754 sjool	389 schooo	204 schok	143 schoolen

Table 1. Most frequent spelling variants for 'school', as suggested by a system built on the principles explained by van Halteren & Oostdijk (2012). The numbers represent the number of instances of the form for which the system suggested the normalized form 'school'.

achol aschol dcholl dnsschool echschool eschool hagol higchool higschool hughschool oschool pschool rschool sachoool sccchoool scchok scchooool scchoot scghoool scgoll schaol schgoool schhhoooll schhlool schhok schhoo schhoolo schhooollll schhoooon schill schjooll schlll schlol schlool schoeonen scholk scholll schollol schollos schooa schoog schoohol schoohoon schoohooon schookll schoolh schoolkl schooloe schoolof schoolollolololllooollolllo schoolp schoolschool schoolse schooltl schoolzl schoolzn schoont schooohl schooola schoooohooool schoooolen schoooolllll schooooohooool schoooooolen schoooooollllll schoooooon schoooooooooon schoooooooooooooooool schooooooooooooooooooool schooop schoow schorel schorn schosol schotel schuool scoll scoolh scooool sggoo sghhoog sgiool sgoil sgoof sgoohool sgookl sgoolc sgooloo sgooollk sgooon sgoooooool sgpool sgvool shcooool shoooool sichool siol sjoooool skoooooooooool sochool sohool sschhool sschoooon sschoooooool sschoot ssssschool svhooll sxcholll sxool vschool wegschool

Figure 1. Spelling variants suggested for only one instance in our data set by a system built on the principles explained by van Halteren & Oostdijk (2012).

In Figure 1, there are more false positives, mostly similarly spelled forms that have been attracted to the cluster by the relative frequency of school (e.g. *shoohooon* is more likely *schoon* ('clean'), and specific types of school (e.g. *higschool* is probably meant to be 'highschool'). All in all, the precision appears to be very high.[13] Table 2 shows the twenty most similar forms in terms of context vectors based on a window of two tokens left and two tokens right, and using a larger data set than in the one used in the previous study (van Halteren, In prep.). We see mostly the same variants, but now in a different order, namely similarity instead of frequency. The order appears to distinguish between intentional variants, such as *sgool* and *schoooool*, which appear to be slightly more distant in context from *school*, and typos, such as *shcool* and *schook*, which are found in much the same contexts as *school*. Notably missing in the top-50 is *sgl*, but closer inspection shows that this is because in 2013 and 2014, there was an extensive discussion about a financial fraud by the director of an institute called SGL, which had repercussions for the measurements underlying Table 2, but not Table 1 as that reflects data up to 2012. Notably added in Table 2 are the forms *scorro/skola* and their variants. These are street language words for school and should therefore be seen as synonyms rather than spelling variants.

One of the more noticeable mechanisms for variation is actually used to attach additional information to the words themselves, namely repetition of individual characters or strings of characters. Such repetition signifies stress, and is a written kind of prosody. When repeating longer substrings, stressed words do become more prone to typos, but given the regular repeating pattern, resolving typos should be relatively easy. Repetition is productive rather than lexicalized, but can be handled as a morphological process, as demonstrated with the Viterstein algorithm (van Halteren & Oostdijk, 2012).

There are a number of other conscious variation mechanisms. First, we see various methods of shortening the text. Shortening is possible, for example, by clipping forms, e.g. *eig* for *eigenlijk* ('in fact'), replacing the full form by an acronym, e.g. *pww* for *proefwerkweek* ('exam week'), vowel deletion, e.g. *gwn* for *gewoon* ('just'), or using rebus-like forms, e.g. *w8* for *wacht* ('wait').

[13] Obviously the data set is far too big to measure recall.

0.7270 shool	0.6848 achool	0.6487 schoowl	0.6015 schoolll	0.5798 schoooooool
0.7082 sschool	0.6828 schhol	0.6485 scholl	0.6014 sqool	0.5738 scoro
0.7018 schhool	0.6808 schoolk	0.6483 schol	0.6012 skoele	0.5618 schooolll
0.6993 scchool	0.6798 scgool	0.6464 sgol	0.5985 schooool	0.5614 schooooooool
0.6952 shcool	0.6679 schooo	0.6308 schooll	0.5983 scorro	0.5591 schoooooool
0.6929 scjool	0.6657 schoool	0.6259 schhoool	0.5981 sgooll	0.5588 schooooll
0.6929 svhool	0.6624 schoo	0.6201 skola	0.5940 skorro	0.5575 scola
0.6905 schiol	0.6596 schoop	0.6104 sgoool	0.5876 schoooll	0.5447 scorroo
0.6875 schook	0.6580 sgool	0.6055 skoool	0.5825 schoooool	0.5440 scoroo
0.6855 sxhool	0.6488 schoolie	0.6046 skolla	0.5820 schoollll	0.5413 scho

Table 2. Most similar forms to the word form 'school', as calculated on the basis of all instances of each form with a text window of two tokens left and two tokens right (van Halteren, in prep.). The numbers represent cosines between the context vectors of school and of the form in question, with the vector dimensions being PMIs between the word form and the context.

Many shortened forms are already quite lexicalized. Shortening is mainly meant for efficiency, but the exact type of shortening used is often indicative of a (confirmed or desired) group membership of the user. Again, generally, users avoid ambiguity, but such avoidance is in the context of the tribe communicated with, and shortened forms may well have other meanings in other contexts, implying that modeling shortening mechanisms, including lexicon formation of lexicalized forms, should be done within the contexts in question.

Another frequent conscious variation is phonetic writing. Here, we also see effects mirroring reduction in speech, in Dutch e.g. *n*-deletions, so that it too can sometimes serve as a shortening mechanism. Phonetic writing is even more an indication of tribe membership and/or user characteristics like the regional background of the user, and can therefore be used to much effect in processing, in the sense that selection of the proper tribe model is more likely to be successful.

There are also instances where spelling variation resulting in deviation from the standard norm is unintentional, and which are traditionally grouped as spelling errors. Here we should distinguish between typographical errors, i.e. errors caused by mismanipulation of the input device, and orthographical[14] errors, i.e. errors caused by lack of knowledge of the correct spelling.[15] How to model typographical errors has been studied extensively, but mostly for traditional text types. The degree to which these errors can be modeled in the Twitter context depends on how regular they are for a specific user, and on the input device used. We may be able to recognize which input device has been used on the basis of the Twitter metadata, or possibly by other effects in spelling and orthography, which could facilitate the recognition of the intended word. For example, there is a higher likelihood of substitution of characters by an adjacent character on the keyboard (e.g. *schook* instead of *school*), but the usefulness of this observation depends on whether a keyboard is used at all, the keyboard layout, and the key selection method.[16] Orthographical errors are more user related, and are often similar to phonetic writing. Here it is the recognition that the user belongs to a specific tribe that can help identify the intended word.

5. Syntax

Considering the previous sections, one might expect the use of syntax in the more professional and "serious" tweets to conform to the norms for standard Dutch, and a more chaotic throwing together of words by the more adventurous users. However, this is in fact unlikely. After all, a reader can be expected to cope with a bit of variation in spelling and orthography, and the author can probably judge what is still comprehensible. To come up with a syntactic structure which is non-standard, but still able to convey the intended message to one's readers is much more difficult, which means that most users can be expected to simply choose (consciously or subconsciously) from their available standard repertoire of syntactic structures.

This assumption is confirmed by an investigation of sets of tweets representing various discussion topics (Oostdijk & van Halteren, 2016). In four topic areas, we took eight related hashtags and, for each hashtag, investigated a random sample of 100 tweets.[17] We (manually) split each tweet into parse units and annotated each parse unit for its syntactic category, e.g. full declarative sentence, elliptic declarative sentence, interrogative sentence, noun phrase, etc., and then examined the distribution of these categories.

[14] This is the term traditionally used in research on spelling errors. Note that our use of the term 'orthography' in this paper is different.

[15] Related are errors against morphology, such as erroneous past participle formation, which we will not analyse here.

[16] An additional complication here is caused by the fact that many of the possible input methods for tweets contain 'user friendly' components adjusting words to what they should be according to the method's statistics, and that users most often do not invest in correcting unwanted adjustments. In such cases, it will be much harder to use knowledge of the input method to reconstruct what the user meant.

[17] For a more detailed analysis, and quantitative information, see the already mentioned Oostdijk & van Halteren (2016).

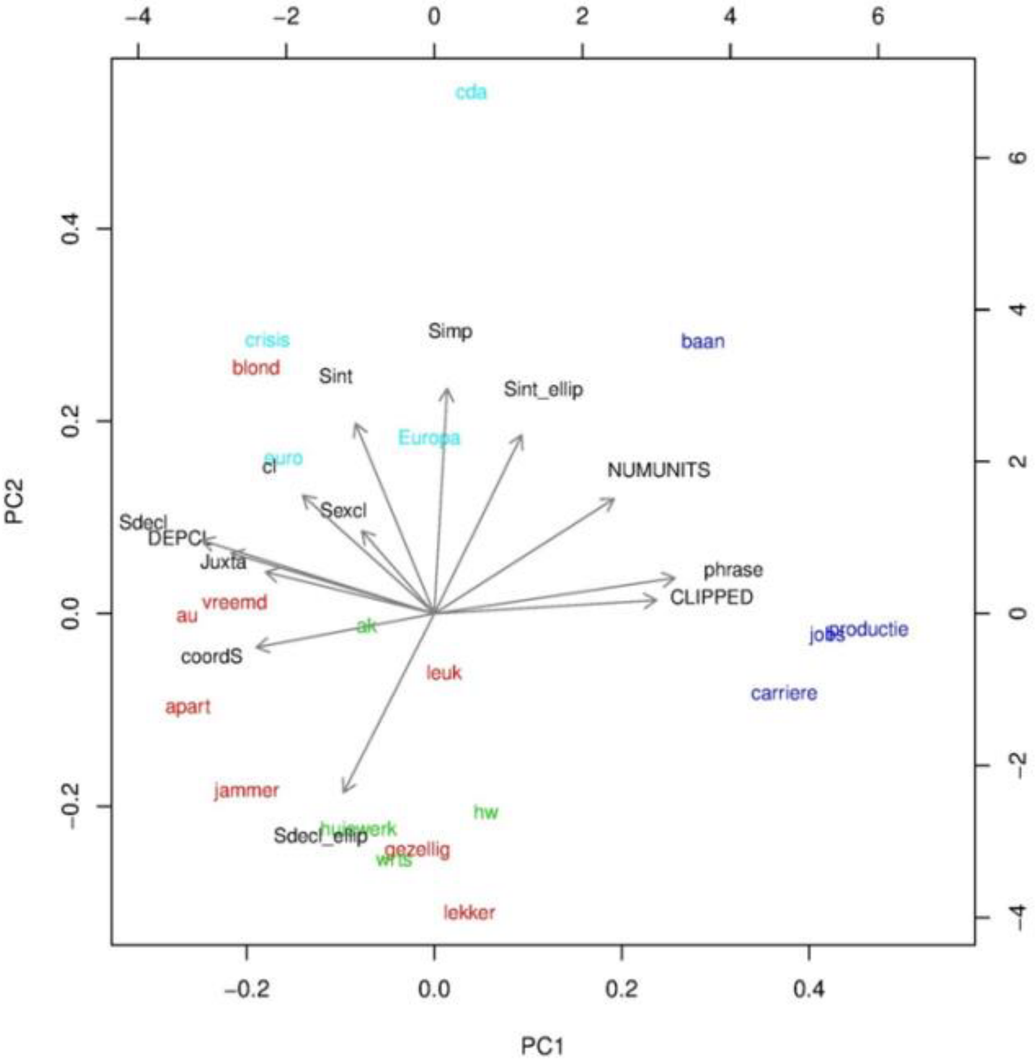


Figure 2: Biplot showing the placement of investigated hashtags and individual annotations in relation to the first two principal components. The hashtags are marked with the cluster colours: 'school' (green), 'employment' (dark blue), 'politics' (light blue), and 'appreciation' (red).

A quick impression of the difference between the clusters can be gleaned from Figure 2, which shows a principal component analysis on the basis of the frequencies of the various parse unit annotations (Oostdijk & van Halteren, 2016).

On the "serious" side of Twitter, we looked at tweets about politics, with hashtags referring to e.g. political parties and political issues. Here, we indeed found mostly full sentences, following standard syntax. To see the extent of variation elsewhere, we targeted tweets where we expected the most severe variation, namely tweets about school, with hashtags referring to e.g. homework and school subjects. Here, we saw a very high frequency of elliptic structures, but most all (over 95%) of the structures encountered were taken from the standard repertoire.[18] The third cluster targeted another extreme, namely the job market, with hashtags referring to e.g. vacancies and career development. Here, we saw a more telegram-like style of communication, trying to pack as much information as possible into the limited space by foregoing sentence structures and replacing them by (sometimes long) sequences of phrases. The phrases, though, followed a standard structure. The final cluster, called "appreciation", was built around hashtags consisting of adjectives expressing an opinion.

In Figure 2, we show the result of a principal component analysis based the frequencies in which the various annotations were assigned in tweets with the various hashtags (Oostdijk & van Halteren, 2016). On the horizontal axis (PC1), we see the distinction between normal clausal structure and phrase stringing, with "employment" favouring the latter and all three other

[18] We do not know whether this observation can be generalized to all tweets, as only tweets with hashtags were included here. We have seen in previous research that tweets without hashtags are more irregular in the sense that they contain more OOV-words (van Halteren & Oostdijk, 2014). We do not know (yet) whether their syntax is also more irregular.

clusters favouring the former. On the vertical axis (PC2), we see the differences in applying the clausal structure, with "politics" mostly adhering to full structures, and apparently also more use of interrogatives and imperatives, and "school" showing much more ellipsis. "appreciation" is spread out over PC2, but is clearly in the clausal camp on PC1. All in all, there are clear differences between topic clusters, but there is also substantial variation within the clusters, implying that widely focused tribal models should already help processing, but that more narrowly focused ones can improve the modeling quality even further.

It would seem that the syntax of tweets can be modeled using much the same methods as for traditional text, at least once the variation in the lower levels of analysis (orthography and spelling; see above) has been accounted for. When using probabilistic methods, however, we would do well to derive probabilities per tribe. Furthermore, such probabilities might also serve to recognize which model should be used for a specific tweet or conversation.

There is one additional complication in the area of syntactic analysis. In some cases,[19] especially when information is forwarded, the text may be clipped, usually marked with an ellipsis sign (…) and a URL. These cases are therefore easy to recognize, but the clipped text is irretrievably lost.[20]

6. Conclusion

In the previous sections, we looked at the wide (and frequent) linguistic variation in the language use on Twitter. Most of this we judge to be intentional, and to be related to the conventions used in the peer group the author belongs to, or would like to belong to, in specific types of communication about specific topics (i.e. what we call *Twitter tribes*). Another source of variation is the author's idiolect, sometimes with clear influences from his/her sociolect. Finally, variation may be caused by mismanipulation of the input device.

All three of these causes are such that we can expect the variation to show a substantial amount of regularity, which means that it can be modeled and that the derived models can be employed in a noisy channel model approach to the normalization of tweets. For various linguistic levels, we have shown the most important processes that constitute the noisy channel. We judge that they can indeed be modeled.

Obviously, we are not the first to suggest a noisy channel model approach. Traditional approaches to (contextual) spelling correction tend to think in terms of noisy channel models (e.g. Dutta et al., 2015) and there is also already experience with applying statistical machine translation techniques for text normalization (e.g. Limsopatham and Collier, 2015). However, we think that this approach is vulnerable because of the heterogeneity of Twitter, and stands to benefit from modeling the patterned variation we see in the language use of tribes.

Taken to its extreme, our proposal would imply that we need to train billions of individual models, which includes finding sufficient training data for each of them. However, as far as we can see, there are gradual rather than radical differences when comparing closely related tribes. We therefore propose to build models for clusters of tribes (which in principle are by themselves also tribes) and use weighted combinations when operating the noisy channel model.

In the near future, we aim to test our proposal. We intend to implement a system that can identify the appropriate tribes (characteristics of author, topic and use situation) for a tweet. In parallel, we will complete our system for linking variant spellings of a word form to a consensus form. [21] Once these are in place, we can evaluate whether tribal modeling indeed outperforms global modeling.

References

Baldwin, T. (Timothy), Cook, P., Lui, M., Mackinlay, A. & Wang, L. (2013). How Noisy Social Media Text, How Diffrnt Social Media Sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013).* Nagoya, Japan, 2013. Pages 356–364.

Baldwin, T. (Timothy), de Marneffe, M., Han, B., Kim, Y., Ritter, A. & Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015)*, Beijing, China.

Baldwin, T. (Tyler) & Li, Y. (2015). An In-depth Analysis of the Effect of Text Normalization in Social Media. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Denver, Colorado. Pages 420-429.

Bryden, J., Funk, S. & Jansen, V. (2013). Word usage mirrors community structure in the online social network Twitter. *EPJ Data Science*, 2013, 2:3.

Dutta, S., Saha, T., Banerjee, S., & Naskar, S.K. (2015). Text normalization in code-mixed social media text. In *Proceedings the IEEE 2nd International Conference om Recent Trends in Information Systems*, 9-11 July 2015, Kolkata, India. Pages 378-382.

Eisenstein, J., O'Connor, B., Smith, N. & Xing, P. (2014). Diffusion of lexical change in social media. *PLOS-ONE*, 9, 11 2014.

Han, B., & Baldwin, T. (Timothy) (2011). Lexical normalisation of short text messages: Makn Sens a #Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.*

[19] In our sample discussed here, as much as 7% of the tweets.

[20] At least within the tweet. It may be present at the URL mentioned, but recovery in such cases is outside the scope of this paper.

[21] The choice for spelling is pragmatic rather than optimal for our goal. We expect that the highest quality gain can be reached in syntax, but a syntactic analysis system is far more difficult to build, if this is even possible as long as the spelling variation is not resolved.

Portland, Oregon, June 19-24, 2011. Pages 368–378.

Kaufman, M., & Kalita, J. (2010). Syntactic normalization of Twitter messages. In *Proceedings of the International conference on natural language processing*, Kharagpur, India.

Limsopatham, N. & Collier, N. (2015). Adapting Phrase-based Machine Translation to Normalise Medical Terms in Social Media Messages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Oostdijk, N. & van Halteren, H. (2016). Twitter Tribal Languages. In: *Handbook of Twitter for Research* (Proceedings CTR2015).

Tjong Kim Sang, E. & van den Bosch, A. (2013). Dealing with Big Data: The case of Twitter. *CLIN Journal* Vol. 3, 121-134.

van Halteren, H. (2015). Metadata Induction on a Dutch Twitter Corpus: Initial phases. *Computational Linguistics in the Netherlands Journal*, Vol. 5. 37-48.

van Halteren, H. (in prep). Word similarity in Dutch tweets. To be submitted to *Computational Linguistics in the Netherlands Journal*, Vol. 6.

van Halteren, H. & Oostdijk, N. (2012) Towards Identifying Normal Forms for Various Word Form Spellings on Twitter, *Computational Linguistics in the Netherlands Journal*, Vol. 2. 2-22.

van Halteren, H. & Oostdijk, N. (2014). Variability in Dutch Tweets. An estimate of the proportion of deviant word tokens. *Journal for Language Technology and Computational Linguistics,* Vol 29(2). 97-123.

van Halteren, H. & Oostdijk, N. (2015). Word Distributions in Dutch Tweets. *Tijdschrift voor Nederlandse Taal- en Letterkunde*. 2015/3. 189-226.

van Halteren, H. & Oostdijk, N. (Submitted). Twitter language model differences between topics and between communities. Submitted to ACL2016.

van Halteren, H. & Speerstra, N. (2014). Gender Recognition on Dutch Tweets. *Computational Linguistics in the Netherlands Journal*, Vol. 4. 171-190.

An Annotation Scheme for the Comparison of Different Genres of Social Media with a Focus on Normalization

Ronja Laarmann-Quante, Stefanie Dipper
Department of Linguistics
Ruhr-University Bochum
ronja.laarmann-quante@ruhr-uni-bochum.de, dipper@linguistics.rub.de

Abstract

Most work on automatic normalization of social media data is restricted to a specific communication medium and often guided by non-comprehensive notions of which phenomena have to be normalized. This paper aims to shed light on the questions (a) what kinds of 'deviations from the standard' can be found in German social media and (b) how these differ across different genres of computer-mediated communication (CMC). To study these issues systematically, we propose a comprehensive annotation scheme which categorizes 'non-standard' (defined as out-of-vocabulary, OOV) tokens of various genres of CMC with a focus on the challenges they pose to automatic normalization. In a pilot study, we achieved a high inter-annotator agreement (Cohen's $\kappa > .8$), which suggests good applicability of the scheme. Primary results indicate that the predominant phenomena are rather diverse across genres and, furthermore, that in some genres, general OOV-tokens, which are not CMC-specific (like named entities or regular non-listed words), play a more dominant role than one might guess at first sight.

Keywords: Social media, annotation scheme, normalization

1. Introduction

The issue of normalizing social media texts has been researched intensively during the last years (see, for example, Eisenstein (2013)). Most approaches, however, focus on exactly one communication medium (e.g. Twitter, SMS) and are usually guided by a rather restricted notion of phenomena that have to be normalized. Statements about the applicability to other genres are often only vague and not further evidenced, as e.g. in Kobus et al. (2008), who deal with normalization of French SMS: "to a large extent, the techniques we present in this paper are also applicable to other types of electronic messages". Similarly, the choice of techniques is not always well justified. Concerning phonetic substitutions in English twitter data, Kaufmann and Kalita (2011) simply state that "[w]e feel that these errors are rare enough that the additional computational complexity required by these models is not justified in this system".

One possibility to overcome such vagueness is to rely solely on statistical methods that do not make any a-priori assumptions (Ling et al., 2013). On the other hand, one can also make use of the fact that different phenomena pose different challenges to normalization and design different "expert modules" to handle them differently (Cotelo et al., 2015). This requires solid qualitative knowledge about the phenomena and their properties, and quantitative knowledge about how prevalent these phenomena are in the texts to be normalized.

In this paper, we propose a comprehensive annotation scheme for German social media texts that categorizes tokens with a focus on the kind of challenges they pose to automatic normalization. Unlike existing categorizations, which we will review in Section 2, our scheme is applicable to all kinds of social media texts and not designed for one genre only. As, for example, Storrer (2013) has shown, the frequency of phenomena related to computer-mediated communication (CMC) varies greatly across genres (chats vs. wiki discussion pages) and context (social vs. professional context). With our annotation scheme, these differences can be assessed quantitatively.

The remainder of this paper is structured as follows: in Section 2, we briefly review existing categorization schemes before presenting our scheme in Section 3. We carried out a pilot study to determine the inter-annotator agreement for our scheme and get first insights into differences between various genres of German social media texts.[1] The procedure and results of this study are presented in Section 4. Section 5 gives a conclusion.

2. Related Work

This section starts with a survey of work on German CMC data, followed by work on other languages.

For German, Bartz et al. (2013) and Dürscheid et al. (2010) propose typologies of phenomena in social media data. Their focus is not on automatic normalization, so the scopes are partly different from our proposed scheme.

Dürscheid et al. (2010) study language use in adolescents' texts produced in the 'new media' compared to texts produced at school. Accordingly, their typology distinguishes categories that are not of primary relevance for normalization, e.g. use of metaphors or colloquial expressions like *geil* 'hot' or *mega* 'mega'. On the other hand, phenomena that behave differently with regard to normalization are subsumed under one category, for instance, inflectives (*knuddel* 'cuddle'), and abbreviations and acronyms (*Compi* for *Computer* 'computer', *WE* for *Wochenende* 'weekend').

Bartz et al. (2013) deal with automatic tokenization and POS-tagging of data from internet-based communication (IBC). Our scheme extends (and modifies) their scheme in that we include phenomena of standard language that are challenging for automatic normalization procedures. This

[1] The data annotated for the pilot study is available at `www.linguistics.ruhr-uni-bochum.de/resources/corpus/cmc_norm`.

concerns word classes that are typically only partially included in word lists or dictionaries, such as foreign words, named entities, or interjections. In German CMC data, interjections and foreign (in particular, English) words are frequent phenomena.

Sidarenka et al. (2013) propose a typology that does consider such kinds of tokens. They investigate how many and which kinds of out-of-vocabulary (OOV) tokens occur in German Twitter data. Hence, their research question is similar to ours with the exception that their research is only directed at Twitter data. This has consequences for the scope of the categorization scheme, though. For instance, action expressions like **aufpluster** 'fluff up' are highly frequent in chat data but seem not to play a role in Twitter messages, hence, they are not featured in their scheme. Furthermore on Twitter, expressions like @*name* are realized as a link to the account of the addressed person. Therefore, one can assume that most addressings are the correctly-spelled usernames. In chat data, on the other hand, addressing is rather informal so that nicknames are often deliberately or accidentally varied by the participants. Furthermore, the @-sign is not always used to indicate a nickname so that they are not trivial to identify.

Sidarenka et al. (2013) propose three main categories which are further subdivided: 1. Limitation of machine-readable dictionaries, 2. Stylistic specifics of text genre, and 3. Spelling deviations. The subcategorization of the first category is very detailed, while the other two categories are only subclassified broadly. Capitalization and word-boundary errors are not covered explicitly, and they define a broad subcategory 'slang' which subsumes different phenomena that we want to keep apart, not only from the point of view of normalization but also for further theoretical questions. For instance, colloquial or dialectal terms (e.g. *nö* 'nope', *bissl* 'a bit') are grouped together with common CMC acronyms (e.g. *LOL, ava*). In addition, spellings which imitate colloquial pronunciation (e.g. *Tach* instead of *Tag* 'day' (salutation), *nen* instead of *einen* 'a') fall under this category as well, but are additionally assigned to the category 'spelling deviations'.

We consider it more adequate to distinguish these cases, not only from the point of view of normalization but also for further theoretical questions.

For other languages, categorization approaches that are similar to Sidarenka et al. (2013) can be found in Cotelo et al. (2015) for Spanish Twitter data and van Halteren and Oostdijk (2014) for Dutch Twitter data. Both lack phenomena that are typical of other CMC genres as addressed above and, not being designed for German, lack categories that would be relevant especially for German data, i.e. umlauts or capitalization of nouns. Besides that, the categorization of Cotelo et al. (2015) is only very broad, for example, it groups together all phenomena that can be detected with help of regular expressions (e.g. hashtags and dates).

van Halteren and Oostdijk (2014) list a larger number of categories but do not provide a useful internal structure. For instance, there is no further sub-division of phenomena that are specifically related to CMC, vs. standard cases that have to be normalized, or OOV tokens that need no further amendment. Furthermore, 'spelling deviations' are sub-classified as lexicalized vs. productive but no further properties are specified, e.g. whether they are phonologically determined, typing errors or otherwise deliberately applied. A particularity of their categorization is that they also considered in-vocabulary tokens which were used in a non-standard way (e.g. with a different meaning in 'street language').

3. Normalization and Annotation Scheme

Our scheme differs in several respects from the schemes described in the previous section. Firstly, it is designed to accommodate phenomena from different genres to allow for comparisons. Secondly, it is supposed to draw a sharp line between tokens that are to be normalized (e.g. spelling deviations) and 'legitimate' OOV tokens, handling both sides equally detailed. We assume that it is a challenge for normalization tools to decide whether an unknown token has to be changed or not, so that such a distinction is meaningful. Furthermore, we want to distinguish different kinds of deviation from the standard, with a focus on differences in the normalization strategy they require. For instance, phonologically-determined spelling deviations and typing errors are different phenomena and, hence, may require different normalization techniques. As many relevant features are already present in the aforementioned works, our scheme can be seen as an extension and restructuring of those. In addition, we designed a tagset representing our categories. Some tags were adopted from the POS-tagging tagsets STTS (Schiller et al., 1999) and its proposed extension for CMC data (Beißwenger et al., 2015b).

Our annotation scheme requires some notion of what 'the standard language' is. In a pilot study (Section 4), we followed the approach by Sidarenka et al. (2013) and used the spell checker Hunspell[2] as a reference but other resources like word lists are conceivable as well. Basically, our scheme is designed to categorize tokens which are 'deviations from the standard' in that they are not captured by such a 'standard language resource', and therefore potentially pose a challenge to further processing tools, and/or are of interest for research about language use in CMC.

The annotation scheme is split in two parts with 46 tags in total. On the one hand, there are tokens which can be normalized sensibly to some standard-language counterpart. These are shown in Table 1. Here, we distinguish three broad categories.

Category KB Firstly, there are keyboard- or fast-typing related phenomena. This accommodates orthographic errors,[3] deviations in capitalization, graphemes only occurring on German keyboards and mistakes in setting word boundaries. These subcategories are grouped together because we believe that they are mostly involuntary, produced by 'slips of the fingers' and only constituting minor deviations from the correct spelling. Some phenomena might as well be intended, e.g. fully uppercased words to signal

[2] `http://hunspell.github.io/`

[3] As the background of this annotation scheme is normalization, the tag names are to be read in the direction 'what has to be done to obtain the correct form', e.g. `ORTH_INS` means that an insertion is necessary.

Cat.	Subcategory	Tag	Description	Example
Keyboard/Fast Typing-Related (KB)	**Orthographic Errors** (ORTH)	ORTH_INS	insertion	*ser* → *sehr* 'very'
		ORTH_DEL	deletion	*niocht* → *nicht* 'not'
		ORTH_REPL	replacement	*unf* → *und* 'and'
		ORTH_SWITCH	permutation	*uach* → *auch* 'also'
		ORTH_PRD	omitted period after standard-language abbreviations	*Mio* → *Mio.* 'million'
		ORTH_OLD	old spelling	*muß* → *muss* 'must'
	Capitalization (CAP)	CAP_FIRST	lowercased noun or name	*schatz* → *Schatz* 'treasure'
		CAP_INNER	some case deviation(s) within the word	*JaHREN* → *Jahren* 'years'
		CAP_INVERSE	letter case inverted	*dU* → *Du* 'you'
		CAP_FULL	full word affected	*SOGAR* → *sogar* 'even'
	Keyboard-Related Variations (VAR)	VAR_UML	umlaut	*muessen* → *müssen* 'must'
		VAR_SS	*ss* → *ß*	*reissen* → *reißen* 'rip'
	Word Boundaries (WB)	WB_SPLIT	missing whitespace	*bittesehr* → *bitte sehr* 'you're welcome'
		WB_MERGE	superfluous whitespace	*schre iben* → *schreiben* 'write'
		WB_SPLITMERGE	whitespace at wrong location	*schona ber*→ *schon aber* 'yes but'
Pronunciation-Related (PR)	Graphical imitation of pronunciation or prosody	COLL_STD	imitation of colloquial but standard-near pronunciation	*nich* → *nicht* 'not', *aba* → *aber* 'but'
		COLL_CONTR	colloquial contraction	*weils* → *weil es* 'because it', *fürn* → *für ein* 'for a'
		COLL_APOSTR	colloquial contraction indicated with apostrophe	*war'n* → *waren* 'were', *auf'n* → *auf den* 'on the'
		ITER	iteration of graphemes or punctuation marks	*sooooo* → *so* 'so', *????* → *?*, *:-)))* → *:-)*
		DIAL	dialectal pronunciation	*wat* → *was* 'what'
		CMC_SPELL	probably intentional deviation which is not primarily phonologically determined but contains a spelling typical of CMC	*Leutz* → *Leute* 'people', *ver3fachte* → *verdreifachte* 'tripled'
Other w/Norm. (OTH_wNORM)	Other kinds of deviations which are to be normalized; most probably not just typing errors but deliberately applied	ABBR_NEO	abbreviation which is not already fixed in standard language	*Bib* → *Bibliothek* 'library', *vllt* → *vielleicht* 'maybe'
		GRAM	deviation in inflection/derivation and grammatical issues	*waschte* → *wusch* 'washed'

Table 1: Tags for tokens which require normalization.

emphasis but from a technical or normalization perspective there is no difference between voluntarily and accidentally holding the shift or caps lock key.

Category PR The second category comprises phenomena related to imitating the pronunciation or prosody of a word. A particularity to note is that we differentiate between imitating a colloquial but standard-near pronunciation and a dialectal one. We assume that the former cases are more frequent, wide-spread and more similar to the standard German counterpart (e.g. *nich* → *nicht* 'not') than dialectal ones, which are sometimes hard to relate to a standard German word on a phonological basis only (e.g. *icke*→ *ich* 'I' in Berlin dialect). Furthermore, we subsume deliberately applied spelling variations under this category as well. Some CMC-specific spellings like *ver3facht* for *verdreifacht* 'tripled' have their origin in the pronunciation as well and even those which are more deviant form the standard language (like *Leutz* for *Leute* 'people') can be seen as a kind of (written) 'dialect' as well.

Category OTH_wNORM Thirdly, there are phenomena which are too complex to be attributed to 'slips of the fingers', and some of them being clearly intentional but hard to predict as there is no clear relation to pronunciation either. However, they can be clearly normalized to a standard German expression. These are grouped under 'other phenomena with normalization' and cover ad-hoc abbrevi-

CAT.	SUBCATEGORY	TAG	DESCRIPTION	EXAMPLE
Regular Vocabulary (LEX)	Regular vocabulary which is not CMC-specific	GAP	standard language word which is not listed in the standard-language resource in question	*Tweet* 'tweet', *emporbringen* 'help forward', *Wortverlaufskurve* 'graph of progression of word'
		REGIO	regional/dialectal expression	*Schrippe* 'bread role' (in Berlin dialect)
		ITJ	interjection	*jo*, *hehe*, *oha*
		NE	named entity	*Yannik*, *Shiva*
		FOREIGN	foreign language	*nine*, *juvare*
Social Media Related (CMC)	**Nicknames** (NICK)	NICK_FULL	full nickname/twitter name, addressed or just mentioned	*@stoeps*, *Erdbeere$*, *marc30*
		NICK_VAR	variation of a nickname, not further analyzed so far	*schtöps*, *erdbäre*, *marc*
	Action Expressions (ACT)	ACT_BEG / ACT_END	asterisk marking the beginning / the end of an action expression	* *wilhelm busch zitier* * 'cite Wilhelm Busch'
		ACT_INFL	inflective	*hinstell* '(to) position', *freu* 'rejoice'
		ACT_ACR	common acronym standing for an action expression	*g* (= *grins*, 'grin'), *lol* (= 'laughing out loud')
		ACT_COMPLEX	action expression without whitespaces	*erleichtertguck* 'look relieved', *neuesuch* 'search new'
	Emoticons/Emojis (EMO)	EMO_ASC	emoticon made of ASCII characters	*:)*; *xD*
		EMO_IMG	coded graphical emoji	*emojiQflushedFace*
	Other CMC-related expressions	ACR	CMC-typical acronyms	*kA* (= *keine Ahnung*, 'no idea'), *wb* (= 'welcome back')
		HST	hashtag	*#dtaclarin14*, *#tatort*
		WEB	URL, domain name, e-mail address	*Fettehenne.info*
		MISC	remaining CMC-specific cases	*nagut50cmlauflaufleine* 'okay 50cm run run leash'
Other w/o Norm. (OTH_woNORM)		DELIB	deliberate creation of a word: ad-hoc neologism, play on word etc.	*leinbruam*, *konfetti* 'confetti' lowercased as adjective
		NONW	no word intended	*sdfsd*
		PUNC	(combination of) punctuation marks	<-
		TECH	technical issues (e.g. incorrect tokenization)	**s**, *51cm* as one token
		UNC	unclear target word	*kommst du zum* ***ct****?* 'do you come to *ct*?'

Table 2: Tags for tokens which do not require normalization.

ations and grammatical/morphological mistakes.[4]

The second part of our annotation scheme is concerned with OOV tokens which are not covered by resources related to 'standard language', such as Hunspell, but still are 'legitimate' in their own right. Hence, these tokens are not to be normalized, although, depending on the aim of the annotation, some could be mapped to standard-language words (e.g. dialectal expressions, CMC acronyms). The full list of categories is given in Table 2. Again, we have three broad categories.

Category LEX Firstly, there are 'regular words' which are simply not listed in the considered standard-language resource but which are not related to CMC per se. Of course, interjections or regional expressions may be more frequent in CMC than in other written genres but they can basically occur everywhere and have existed before the rise of CMC. We also subsume foreign words under this category because these are 'regular words' as well, simply not

[4] The example *waschte* → *wusch* 'washed' is taken from Dürscheid et al. (2010).

in modern German.

Category CMC Secondly, we have a CMC-specific category. This comprises all phenomena that are clearly products of computer-mediated communication. Our subcategories are more comprehensive and fine-grained, though, than those in the existing typologies reviewed in Section 2. On the one hand, this is because we do not focus on one specific genre of CMC, and on the other hand, because we want to capture the different challenges for normalization. That is why, for example, we distinguish full nicknames and their variations, and different kinds of action expressions.

Action expressions can be inflectives (*grins* 'grin'), a sequence of words without whitespace (*immernochnichtfassenkann* 'still not be able to comprehend'), or an acronym standing for an expression (*lol*). Furthermore, they can consist of a sequence of words including spaces like (* *Wilhelm Busch zitier* * 'cite Wilhelm Busch'). In the data we used for our pilot study (section 4.), the star signs and each word in such expressions were analyzed as individual tokens so that the inflective (*zitier* 'cite') would be the only actual OOV token here.

Category OTH_woNORM Our third category captures all other OOV tokens which are legitimate the way they are but neither 'regular German words' nor are particular to CMC.

Tokens can carry multiple tags. If so, tags affecting the whole word are applied first, then those which only affect parts of the word from left to right. The superordinate category precedes the tag with a colon. Here are some examples:

ORIG.	TAGS	NORM.
`sone`	`PR:COLL_CONTR`	`so eine`
`juhuuu`	`LEX:ITJ,PR:ITER`	`juhu`

`zeugniss`	`KB:CAP_FIRST,`
	`KB:ORTH_DEL`
		`Zeugnis`

4. Pilot Study

In our pilot study, we wanted to test the applicability of our annotation scheme by assessing the inter-annotator agreement and also get some first insights in what kinds of differences there are between different genres of CMC.

4.1. Data

The data we used were the CMC training data provided for the currently running shared task "EmpiriST 2015 shared task on automatic linguistic annotation of computer-mediated communication/social media"[5] which aims at automatic tokenization and POS-tagging of German CMC and web data. These consisted of 5106 tokens in total, distributed across different genres as follows:[6]

- **Tweets**: 1,163 tokens; 153 tokens taken from the Twitter channel of an academy project; 1,010 tokens taken from the Twitter channel of a lecturer in German Linguistics, used for discussions with students accompanying a university class.
- **Social Chat**: 1,100 tokens, taken from the Dortmund Chat Corpus[7].
- **Professional Chat**: 1,006 tokens, taken from the Dortmund Chat Corpus.
- **Wikipedia Talk Pages**: 925 tokens, taken from two talk pages of the German Wikipedia.
- **WhatsApp Conversations**: 554 tokens, taken from the data set collected by the project "WhatsApp, Deutschland?"[8].
- **Blog Comments**: 358 tokens, taken from weblogs under CC license.

We used the manually-tokenized data provided by the shared task. These followed the tokenization guidelines for CMC data in Beißwenger et al. (2015a).

4.2. Procedure

As a reference for 'German standard language', we used the generic spell checker *Hunspell*.[9] Our basic idea was to annotate all tokens which were not recognized as valid German words by the spell checker. We considered Hunspell a suitable reference because it is very common in open source tools such as Mozilla Firefox and LibreOffice, supports compounding and complex morphology and contains large dictionaries including frequent proper names and standard abbreviations.

In a preprocessing step, we automatically marked all tokens which were not recognized by Hunspell. We further marked all fully-uppercased tokens (which Hunspell incorrectly accepts in general).[10]

Furthermore, we automatically pre-annotated asterisks as potential boundaries of action expressions, single words between asterisks as potential inflectives, all single-letter tokens as potential acronyms, and all tokens that were un-

[5] `http://sites.google.com/site/empirist2015/`

[6] Description taken from the *readme* file distributed with the data.

[7] `http://www.chatkorpus.tu-dortmund.de`

[8] `http://www.whatsup-deutschland.de/`

[9] v.1.3.2; `http://hunspell.sourceforge.net/`, accessed with a Python 3.4 script via the Python module *hunspell*, v.0.3.3, `https://pypi.python.org/pypi/hunspell`; German dictionary files: v.20131206.

[10] Hunspell also tolerates capitalized words in general (with only the first letter in upper case) because this spelling is required for sentence-initial words. In contrast to fully-uppercased tokens, we did not premark such cases and, hence, they are not annotated in our corpus.
For some reason, Hunspell does not recognize standard contractions with *'s* for *es* 'it', as in *wär's* for *wär es* 'were it'. These cases were only pre-marked if Hunspell did not recognize the base part of such forms (*wär* in the example).
Similarly, Hunspell if envoked by Python does not recognize single punctuation marks. These cases were ignored as well.

known to Hunspell but identical to a clear nickname as nicknames.[11]

The manual annotation and normalization was carried out by the two authors of this paper on the basis of these pre-annotations. Only words that were marked by Hunspell and the pre-annotations were annotated, so for this pilot study, potential real-word errors and purely grammatical errors were ignored.[12] Furthermore, we skipped manual normalization of tokens annotated (exclusively) as `CAP_FIRST` because it is trivial. We used LibreOffice as annotation tool, with pre-defined drop-down menus containing all tags as well as some selected combinations of tags.

4.3. Inter-annotator agreement

We measured inter-annotator agreement between the two authors by (raw) percent agreement and Cohen's κ on all annotations.[13] If a token was annotated by several tags, the token was multiplied so that each token was assigned one tag. If the annotations of both annotators were complex, the best possible alignment between the annotations was chosen. For instance, for the word *reee*, Annotator 1 had (correctly) chosen `CMC:ACR` and Annotator 2 `CMC:MISC`, both annotated `PR:ITER`.

ORIG.	ANNO-1	ANNO-2
reee	CMC:ACR,PR:ITER	CMC:MISC,PR:ITER

For computing agreement, these annotations are converted as follows:

ORIG.	ANNO-1	ANNO-2
reee	CMC:ACR	CMC:MISC
reee	PR:ITER	PR:ITER

Table 3 lists the agreement figures for all three categorial levels and by genre.[14] Following the interpretation by Landis and Koch (1977), the table shows that we achieve "almost perfect agreement" ($\kappa > .8$) for most cases.

Surprisingly, agreement on the Wikipedia Talk Pages is considerably lower. The main reason is that one of the Wikipedia discussions evolves around the correct German term for "songwriter". The term used in the Wikipedia article under discussion is *Liedschreiber*, an uncommon term, which is the literal translation of the English term *songwriter*. One of the annotators marked this word as `LEX:GAP`, assuming that it is actually a standard language word. The other annotator marked it as `OTH_woNORM:DELIB`, i.e. an ad-hoc neologism. There are six instances of *Liedschreiber* in the corpus, and four of *Liederschreiber(ei)*, meaning that this term alone contributes 15% to the disagreement on the Wikipedia Talk Pages.

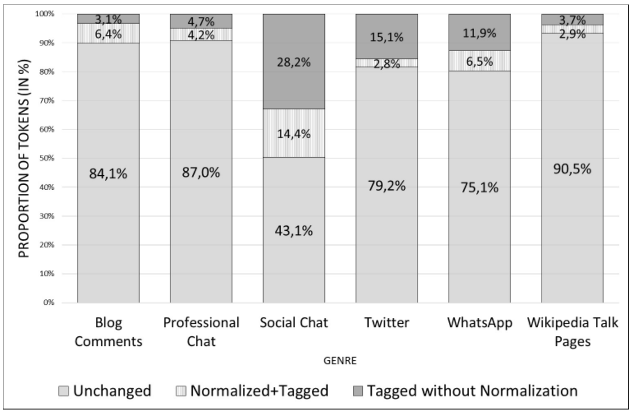


Figure 1: Proportions of tagged and normalized tokens by genre.

4.4. Results

For our pilot analysis of differences across CMC genres, we used the annotations by one of the authors. Figure 1 shows the proportion of normalized and annotated tokens for each genre. The comparison reveals that the number of unannotated tokens that were recognized by Hunspell (*unchanged*) varies considerably across genres and also depending on the setting (e.g. Social vs. Professional Chat). However, in some genres (e.g. Twitter and WhatsApp on the one hand, and Blog Comments and Professional Chats on the other hand) the proportion of OOV tokens is pretty similar. In addition, one can see that especially in the Social Chat, Twitter and WhatsApp, there are considerably fewer OOV tokens that actually have to be normalized (*normalized+tagged*) than those which are 'legitimate' the way they are (*tagged without normalization*).

Looking at how the main categories are distributed, Figure 2 reveals that in these three genres (Social Chat, Twitter and WhatsApp) the most frequent category are CMC-specific tokens (`CMC`). The particular phenomenon that plays the greatest role, however, varies considerably with the genre, as illustrated by Table 4. The table lists the three top-frequent tags for each genre, along with the total number of annotations (#Annos) and the number of attested tag types (#Tags) by genre. For example, in the Blog Comments, 34 tags were annotated, which are instances of 12 different tag types.[15] The table shows that the three genres are characterized by highly typical expressions such as

[11] Clear nicknames are all author names occurring in `<posting>` tags in the Chat data.

[12] In a manual inspection of the tokens recognized by Hunspell across all genres, we found 28 undetected case deviations (e.g. *wagen* 'dare' for *Wagen* 'wagon'), 18 instances where an existing but not intended verb form was used (e.g. *hab* (imparative) for *habe* (1st Pers. Sg. Pres. of *haben*, 'have')), 3 of them in action expressions, 7 undetected nicknames (e.g. *Beere* 'berry' for *Erdbeere$*), 2 undetected non-standard abbreviations (*v. t.* for *Verb transitiv* 'verb transitive', one agreement error (*einen SMS* for *eine SMS* 'one SMS') and one other real-word error (*haste* 1st Pers. Sg. Pres. of 'to hurry' for *hast du* 'do you have'). In total, these real-word errors make up 1.5% of the tokens recognized by Hunspell.

[13] For computing agreement, we used the software tool R and the package 'irr', `https://cran.r-project.org/web/packages/irr/`.

[14] The two Twitter files have been merged as one of them is too small to be analyzed separately. The same holds true for the two Wikipedia Talk Pages.

[15] The raw figures in Table 4 differ slightly from the ones in Table 3. This is due to the fact that Table 4 only reports figures from the first annotator whereas Table 3 shows figures from both

GENRE	SIZE	TAG		SUBCAT.		CATEGORY		NORM.	
		perc.	κ	perc.	κ	perc.	κ	perc.	κ
All	1094	83.00	82.13	85.19	83.90	88.12	83.53	92.60	84.89
Blog Comments	36	86.11	82.84	88.89	84.91	91.67	86.86	94.44	93.48
Prof. Chat	98	87.76	86.72	87.76	86.48	88.78	85.54	88.78	82.23
Social Chat	544	84.74	83.61	86.40	84.34	90.07	84.79	93.75	86.82
Twitter	226	84.51	82.80	85.84	84.24	88.05	82.09	90.27	63.49
WhatsApp	123	84.55	81.33	87.80	83.32	87.80	82.60	94.31	90.02
Wikip. Talk Pages	67	52.24	49.88	62.69	59.30	70.15	63.66	92.54	87.80

Table 3: Inter-annotator agreement (percent agreement and Cohen's κ) for all annotations ("All", 1094 annotations) and by genre. Agreement results are given for individual tags ("Tag"), subcategories ("Subcat."), and main categories ("Category"), and for normalizations ("Norm.").

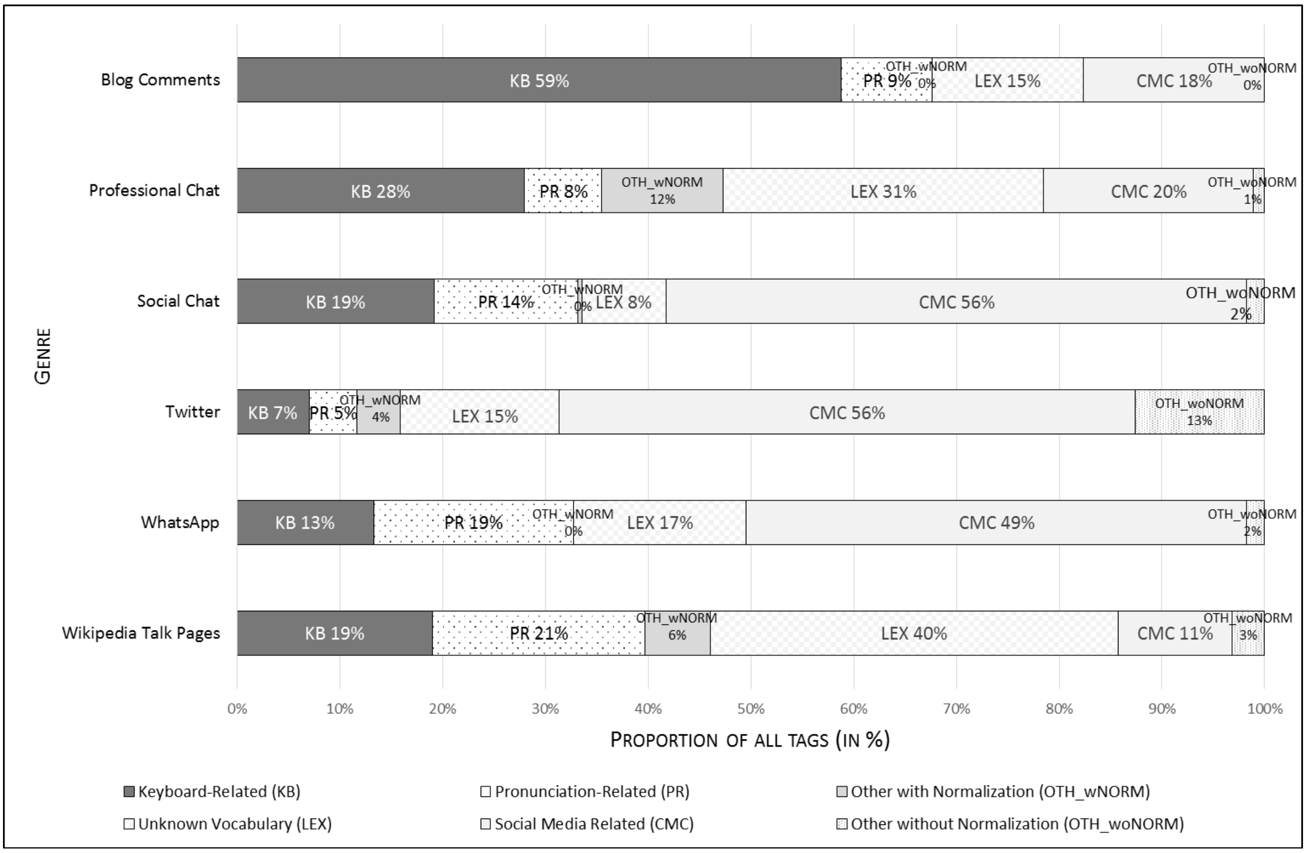


Figure 2: Distributions of main categories by genre.

hash tags (Twitter), nick names (Social Chat, Twitter), or emoticons (WhatsApp).

On the other hand, the Wikipedia discussion, Professional Chat and Blog Comments are rather characterized by keyboard-related deviations (KB) and unknown regular vocabulary (LEX), see Figure 2. However, one can also find indications of a deliberately informal style that is said to be typical of CMC in general, like emoticons and the imitation of pronunciation. The most frequent individual tags (Table 4) are rather diverse here and may be the result of the topic or idiosyncracies of the authors. For instance, the most frequent tag of the Blog Comments are fully uppercased tokens, caused by one longer post (most probably due to an incorrectly activated Caps Lock key).

Our results indicate that interesting differences between genres of CMC do exist in German and that our annotation scheme is able to capture these. In particular, these differences can have an important impact on normalization procedures. In some genres, OOV tokens are predominantly CMC-specific tokens, unrelated to any standard language tokens, whereas in other genres, unknown 'standard language' words play a much larger role and should not be underestimated. Generally, the comparatively high number of tokens that have to be 'recognized' as legitimate OOV

annotators. The second annotator in general tended to annotate more tags than the first.

Genre	#Annos/#Tags	Most frequent tags (%)	
Blog Comments	34 / 12	KB:CAP_FULL	41.2
		CMC:EMO_ASC	14.7
		LEX:NE	8.8
Professional Chat	93 / 21	KB:CAP_FIRST	11.8
		LEX:FOREIGN	11.8
		OTH_wNORM:ABBR_NEO	11.8
Social Chat	510 / 31	KB:CAP_FIRST	14.1
		CMC:NICK_VAR	12.0
		CMC:NICK_FULL	9.4
Twitter	214 / 25	CMC:HST	22.0
		CMC:NICK_FULL	16.8
		CMC:EMO_ASC	11.7
WhatsApp	113 / 13	CMC:EMO_ASC	30.1
		CMC:EMO_IMG	18.6
		PR:ITER	15.0
Wiki Talk Pages	63 / 19	LEX:GAP	22.2
		PR:COLL_STD	12.7
		LEX:FOREIGN	12.7

Table 4: Most frequent tags by genre.

tokens rather than being normalized indicates that normalization procedures should not be too greedy.

5. Conclusion and Outlook

We presented a comprehensive annotation scheme for different genres of German CMC data that captures 'deviations from standard language' that can be relevant for (automatic) normalization. Our annotations are fine-grained enough to be of interest for linguists of other fields as well, for instance, to compare specific CMC-related features across genres. Furthermore, the scheme can be used for qualitative and more detailed evaluations of different normalization techniques, in that it allows statements about which phenomena are handled better than others instead of only giving an overall score for the normalization success. Up to now, the scheme has been applied to a rather small set of CMC data, resulting in very high inter-annotator agreement. It would be desirable to evaluate the scheme with more annotators and also in direct comparison with data annotated by Sidarenka et al. (2013) and Bartz et al. (2013).

6. References

Bartz, T., Beißwenger, M., and Storrer, A. (2013). Optimierung des Stuttgart-Tübingen-Tagset für die linguistische Annotation von Korpora zur internetbasierten Kommunikation: Phänomene, Herausforderungen, Erweiterungsvorschläge. *JLCL*, 28(1):157–198.

Beißwenger, M., Bartsch, S., Evert, S., and Würzner, K.-M. (2015a). Richtlinie für die manuelle Tokenisierung von Sprachdaten aus Genres internetbasierter Kommunikation. Guideline document from the Empirikom shared task on automatic linguistic annotation of internet-based communication (EmpiriST 2015). `http://sites.google.com/site/empirist2015/`.

Beißwenger, M., Bartz, T., Storrer, A., and Westpfahl, S. (2015b). Tagset und Richtlinie für das Part-of-Speech-Tagging von Sprachdaten aus Genres internetbasierter Kommunikation. Guideline document from the Empirikom shared task on automatic linguistic annotation of internet-based communication (EmpiriST 2015). `http://sites.google.com/site/empirist2015/`.

Cotelo, J. M., Cruz, F. L., Troyano, J., and Ortega, F. J. (2015). A modular approach for lexical normalization applied to Spanish tweets. *Expert Systems with Applications*, 42(10):4743–4754.

Dürscheid, C., Wagner, F., and Brommer, S. (2010). *Wie Jugendliche schreiben: Schreibkompetenz und neue Medien*. De Gruyter, Berlin.

Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 359–369.

Kaufmann, M. and Kalita, J. (2011). Syntactic normalization of Twitter messages. In *Proceedings of the International Conference on Natural Language Processing (ICON 2011)*, pages 149–158.

Kobus, C., Yvon, F., and Damnati, G. (2008). Normalizing SMS: Are two metaphors better than one? In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08), volume 1*, pages 441–448.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1).

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2013). Paraphrasing 4 microblog normalization. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 73–84.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universities of Stuttgart and Tübingen.

Sidarenka, U., Scheffler, T., and Stede, M. (2013). Rule-based normalization of German Twitter messages. In *Proceedings of the GSCL Workshop Verarbeitung und Annotation von Sprachdaten aus Genres internetbasierter Kommunikation*.

Storrer, A. (2013). Sprachstil und Sprachvariation in sozialen Netzwerken. In Barbara Frank-Job, et al., editors, *Die Dynamik sozialer und sprachlicher Netzwerke: Konzepte, Methoden und empirische Untersuchungen an Beispielen des WWW*, pages 331–366. VS Verlag für Sozialwissenschaften, Wiesbaden.

van Halteren, H. and Oostdijk, N. (2014). Variability in Dutch Tweets. An estimate of the proportion of deviant word tokens. *JLCL*, 29(2):97–123.

Dialog Act Recognition for Twitter Conversations

Tatjana Scheffler and Elina Zarisheva
Department of Linguistics   Hasso-Plattner-Institute
University of Potsdam, Germany
tatjana.scheffler@uni-potsdam.de

Abstract

In this paper, we present our approach to dialog act classification for German Twitter conversations. In contrast to previous work, we took the entire conversation context into account and classified individual segments within tweets (a tweet can contain more than one segment). In addition, we used fine-grained dialog act annotations with a taxonomy of 56 categories. We trained three classifiers with different feature sets. The best results are achieved with CRF sequence taggers. For the full DA taxonomy, we achieved an f-measure of up to 0.31, for the reduced taxonomy (12 DAs), up to 0.51, and minimal taxonomy (8 DAs), 0.72, showing that dialog act recognition on Twitter conversations is quite reliable for small taxonomies. The results improve on previous work on speech act classification for social media posts. The improvement is due to two factors: (i) Our classifiers explicitly model the sequential structure of conversations, whereas previous approaches classify individual social media posts without taking dialog structure into account. (ii) We segmented tweets into utterances first (segmentation is not a part of this work), while all previous approaches assign exactly one speech act to a post. In our corpus, over 30% of tweets consist of several speech acts.

Keywords: dialog act recognition, speech acts, Twitter, dialog, German

1. Introduction

The first analyses of social media data typically target individual posts when trying to apply natural language processing algorithms such as normalization, POS-tagging, parsing, etc. When the social context has been considered, this has usually been done in the shape of metadata such as user networks. In the meantime, the structure with which users themselves most often interact, the conversation, has not been the focus of many previous analyses. For some social media, such as Twitter, this is mostly due to the fact that obtaining full conversations is not easy through standard API access. If the social network structure is considered in the analysis of Twitter data, this is done by considering relations between users (following, retweet or reply networks) and not between individual tweets.

In contrast, in this work, we study tweets within their native, conversational context. Our aim is to gain insights into the structure of Twitter conversations. In this paper, we address dialog act analysis, as a first step to characterize the structure of dialogs on Twitter. We have developed an automatic dialog act recognition system trained on a set of hand-annotated German Twitter conversations. The dialog act (DA, a reinterpretation of speech act (Searle, 1969)) of an utterance characterizes its *function* in the conversation, independently of its content or topic. Typical dialog acts are INFORM (a statement), QUESTION, AGREEMENT, etc. Predicting dialog acts for social media utterances can be of great use to downstream applications such as the identification of influencers (who may give answers or opinions) or sentiment, and is a prerequisite for automatic human-computer interaction on social media through chat or customer service bots. Further, DAs give cues about the structure and purpose of social media conversations, such as whether the conversation is an instance of an information exchange, argumentation, social chit-chat, negotiation, etc. This is of great value since social media conversations are of very varied types.

In this paper, we present an algorithm to predict fine-grained dialog act sequences on German Twitter conversations, out of a set of 56 DAs. Our approach is adapted from previous work on human-human spoken conversation. We show that despite their brevity, tweets are often composed of more than one communicative intention (DA). Further, by comparing different classification approaches we demonstrate that the sequential nature of the dialog structure is relevant to determining a tweet's DA: sequence modelling approaches such as Hidden Markov Models and Conditional Random Fields work better than mere classifications. The recognition results for the best approach (CRF) are similar to reported results for other corpora on a significantly simplified tag set of 8 DAs. However, we run into a serious shortage of annotated data for the fine-grained classification of 56 DAs. We conclude by giving suggestions for future work.

2. Related Work

Dialog acts have been a main focus of many analyses of human-human dialogs (as well as human-machine interaction, which is not analysed here). Several approaches to DA recognition have been proposed for different data sets. (Stolcke et al., 2000) use a Hidden Markov Model (HMM) to predict a simplified DAMSL tag set of 42 dialog acts on spontaneous telephone speech. They report a very high accuracy of 71% for a combination of several different predictors, using structural, lexical and prosodic features. In contrast, (Ang et al., 2005) use a MaxEnt classifier over a small set of 5 broad DA classes. They report an overall classification accuracy of 81% on gold segments and based on lexical and prosodic features, which is only marginally improved by adding sequence information. This corresponds to an agreement with the gold standard annotation of Cohen's $\kappa = 0.70$.

For social media data, (Forsyth and Martell, 2007) built a dialog act recognizer for chat messages with a custom-made schema of 15 dialog acts. They consider each turn to correspond to only one DA, even though they note that several acts can appear within one turn in their data. For Twitter data, the earliest work is (Ritter et al., 2010), who use unsupervised learning to extract "dialog act" functions from Twitter data. Their system learns 8 DAs that were manually inspected and received labels such as STATUS, QUESTION, REACTION, COMMENT, etc. They also obtain an informative transition model between DAs from their data. In contrast, (Zhang et al., 2011) built a supervised DA recognition system for 5 broad speech acts (STATEMENT, QUESTION, SUGGESTION, COMMENT, MISC), using 8613 handannotated tweets. They used an SVM model with linear kernel, and report an average F1 score of 0.695 for their full feature set, which includes only lexical features and does not take conversation structure or context into account. In newer work, (Arguello and Shaffer, 2015) trained independent binary classifiers to predict 7 DAs for MOOC forum posts. They used logistic regression with different feature sets, including lexical, sentiment and structural (e.g., confidence values of previous DAs, author and time) features. For the full feature set, they achieve average precision values of around 0.65 for each DA (values between 0.15 and 0.82).

Finally, all the previous work on DA classification in social media assign exactly one DA to each post, though it is clear that many social media posts contain several distinct communicative intentions, either in parallel or subsequently.

3. Data

We propose a supervised approach to DA classification. In this section, we introduce the corpus, the designed DA taxonomy, and the annotation process used. We consider *conversations* on Twitter as the basic structures of analysis. Conversations are created when a user replies to a previous existing tweet. In Twitter, as opposed to face-to-face spoken conversation, multiple replies to a tweet can lead to a tree structure that is not limited in depth and width. In addition, the number of participants in a Twitter conversation is not limited. Nevertheless, we often refer to such conversations broadly as dialogs. On Twitter, up to 40% of all tweets are part of conversations (Scheffler, 2014; Honeycutt and Herring, 2009), and conversations can consist of up to hundreds of tweets.

It is difficult to collect complete conversations through the standard Twitter API access, since it allows only random subsets of tweets to be collected. For 'smaller' languages other than English, it is, however, possible to collect near-complete snapshots of tweets over a time period, from which near-complete conversations can be reassembled (Scheffler, 2014). In this paper, we used data that was already collected within the BMBF project *Analysis of Discourses in Social Media*[1]. The dataset is composed of 1566 German tweets (172 dialogs), collected using keywords from the topic *Energiewende* ('new energies/energy transition') in 2013.

After cleaning non-German and blank tweets plus their dependents, 1234 tweets (157 dialogs) remain. We have manually annotated these conversations with dialog acts with the help of minimally-trained students. More detail on the annotation process, the data and the conversion into the gold standard can be found in (Zarisheva and Scheffler, 2015). We employed an adapted version of the multi-purpose DIT++-schema (Bunt et al., 2010), an ISO-compatible, topic-independent dialog act taxonomy for human or human-machine dialogs. The full taxonomy used here contains 51 fine-grained DAs. We added an extra "0" label for usernames at the beginning of reply tweets that are used to address the interlocutor.[2] In order to allow for better comparison with previous work on social media (which uses very broad DA categories), we have introduced smaller taxonomies by merging related DAs (for example, different subtypes of questions), yielding a reduced (12 DAs + "0"-tag) and a minimal (8 DAs + "0"-tag) dialog act taxonomy. The annotators were asked to carry out segmentation of communicative intentions simultaneously with DA classification. Each tweet was seen by 3 annotators. There was a very good agreement on the segmentation (Fleiss' multi-π=0.89, (Artstein and Poesio, 2008)). For the DA labelling, the agreement between 3 annotators rose with decreasing taxonomy size (full schema: π=0.56; reduced: π=0.65; minimal: π= 0.78). This reflects in part the lack of training and flawed understanding of the guidelines, and in part the difficulty of making fine-grained distinctions between communicative intentions based on short, noisy tweets. The annotations were merged by a variant of majority vote and manually corrected, to obtain the gold standard forming the basis of this work (Zarisheva and Scheffler, 2015).

In contrast to previous work on the classification of communicative intentions in social media, we do not assume that each post expresses only one intention (DA). Our manual annotation has shown that despite the brevity of most tweets, about 35% of the tweets carry out two or more subsequent speech acts. A typical example is found in (1). We therefore performed DA classification for smaller, utterance-sized segments within tweets, which allows for a better characterization of the communicative intent of a social media contribution. On the other hand, this leaves very little data for classification: the median segment (not counting "0" segments) only consists of 6-7 word tokens.

(1) True, unfortunately. | But what about the realization
of high solar activity in the 70s and 80s?
AGREEMENT | SETQUESTION

As expected for DA annotation, the distribution of segments over DA classes is very uneven. Table 1 shows the DA unigrams for the reduced taxonomy. The 0-label is used only for tweet-initial usernames that are used to address the tweet. Since these segments constitute 40% of all segments, and are easily distinguished, we excluded these segments from any classifiers or evaluation, so as not to bias the results unduly. The most frequent 'true' DA is INFORM, with about 40% of the remaining segments. Some types of DAs, such as PERSONAL COMMUNICATION MANAGEMENT (PCM, = self-corrections) or OTHER COMMUNICATION MANAGEMENT (OCM, = corrections) are very rare.

[1] http://www.social-media-analytics.org/

[2] The original taxonomy had 56 DAs. However, since some DAs were never assigned by annotators, only 51, plus the "0" tag, were used in this work. The full schema is provided in the appendix.

DA name	# segments	%
0 ("@user")	1144	41.0
true DAs:		
INFORM	664	40.3
QUESTION	255	15.5
DISAGREEMENT	148	9.0
AGREEMENT	146	8.9
DSM	130	7.9
ADF	127	7.7
SOCIAL	73	4.4
INFORMATION PROVIDING	72	4.4
PCM	14	0.9
OTHER	11	0.7
OCM	6	0.4
INFORMATION TRANSFER	0	0
Sum	**2790**	

Table 1: DA unigrams for the reduced DA taxonomy. Percentages are computed excluding the 0-segments.

4. Dialog Act Recognition

We chose a relatively standard classification approach to DA recognition. Even though it is possible to carry out segmentation and DA labelling in tandem (e.g., through a token-based sequence labelling approach), we do not pursue this in this work. We see utterance segmentation of tweets as a separate, complex task that has been gaining some attention in the literature. Thus, all experiments for DA recognition reported in this work are classifications based on gold segments.

In our work, a segment can have one and only one DA label. Since we want to use the dialog structure to inform the DA labelling (e.g., an ANSWER is much more probable when preceded by a QUESTION), we need to sequentialize the Twitter conversations, which are multilogs with possibly many parallel answers to an individual tweet. Figure 1 represents a Twitter conversation. It has two branches, as the root tweet receives two replies. For our work, we broke the tree into individual threads at each internal node, splitting up the conversation in Figure 1 into a thread with ⟨ tweet1, tweet2, tweet3 ⟩ and a second thread ⟨ tweet1, tweet4, tweet5 ⟩. Thus, we observed the root tweet two times.

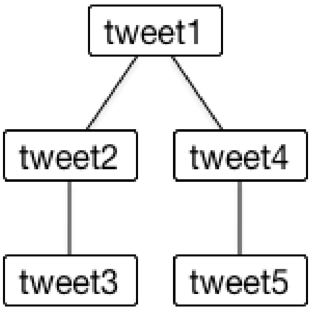


Figure 1: A conversation on Twitter. Edges represent reply relations.

4.1. Features/Algorithms

Among the various classification algorithms (Support Vector Machines, decision trees, etc.) that are commonly used for multiclass classification tasks, and as we pay attention to the links between tweets, we choose those that pay attention to the sequences of states: Hidden Markov Models (HMM) and Conditional Random Fields (CRF). We compared the results provided by both algorithms with each other because HMMs take into account only the previous state of the chain whereas CRFs observe the whole chain at a time.

Because we have a rather small corpus, we used 10-fold cross validation to train 3 different supervised learning algorithms (CRF, HMM with Multinomial and with Gaussian distributions), with different feature sets (see below). For the cross-validation, we distributed complete conversations over the 10 folds, keeping the reply links between tweets intact. This is necessary to maintain the sequence of dialog acts, and it guarantees that the threads in the evaluation fold are completely new and unseen.[3] Since the conversations have very different sizes, the folds have the same number of conversations but the difference in the number of tweets can be enormous. We will see that this has some impact on the recognition results.

In this section, we introduce the features that we used to train the models. All features are lexical or structural and informed by features commonly used in previous work on DA recognition. The first feature set is called the user-defined set (UD). It consists of 12 lexical and structural features, defined over the entire segment under consideration:

- the number of words in the segment;
- (binary feature, 0/1) whether the author of the given tweet is the root author of the conversation;
- the total number of segments in the given tweet;
- (0/1) given segment has a link;
- (0/1) given segment has a question mark;
- (0/1) given segment has a question word;
- (0/1) given segment has an exclamation mark;
- (0/1) given segment has a hashtag;
- (0/1) given segment has an emoticon;
- (0/1) first token in the given segment is a verb;
- (0/1) first token in the given segment is an imperative;

[3] An alternative would be to distribute individual conversation threads over the folds, leading to more evenly sized folds, but also some potential overlap between the data in the training and evaluation parts.

- (0/1) given segment contains the word *oder* ('or').

Additionally, we used two different feature sets modelling the lexical composition of the segment. For each dialog act, we implemented a basic language model (LM TOPX) that contains the top X stems for that DA according to their tf-idf value, where we can choose X manually from 0 to 100. In our work, we used only unigrams, i.e., we observed which words are more frequent in a particular DA. We did not consider bigrams ($n = 2$) or higher because our training set is too small.

Recently, deep learning-based word embeddings have gained importance as semantic context representations in computational linguistics. Word embeddings are vector representations of a word that allow less sparse language models (since large corpora can be reduced to word embedding vectors of only a few hundred dimensions). Here, we used pre-calculated embeddings for the German language made by the Polyglot project (Al-Rfou et al., 2013). In Polyglot, the context window is set to four (two previous words and two following words). The number of dimensions that is used there is 64.

We implemented each feature set with the three classifiers. For CRFs, this is straightforward. For HMMs, the state space is discrete but the observations can be either discrete (multinomial distribution) or continuous (Gaussian distribution). Because word embeddings consist of M-dimensional vectors, we were not able to use multinomial HMMs. This feature set was only used in the *HMM* with Gaussian distribution.

Finally, as an alternative or addition to the automatically predicted DAs, we defined several rules that assign DA labels directly to segments. Since we introduced the "0" DA for all segments that have only usernames in it, we added a rule assigning this tag. Other rules were formed after determining specific patterns in the gold standard annotation. For example, utterances tagged with CHOICEQUESTION in 90% of cases contain the word *oder* ('or') and a question mark. We checked every segment on appearance of these two characteristics, and if both are found within one segment, we assign the CHOICEQUESTION label to this particular segment. We did not find any other specific patterns for DAs from the full DA taxonomy. This may be due to the small size of our data set.

For both the reduced and minimal DA taxonomies, we introduced two more rules. The first rule is a generalization for all types of questions since in these two taxonomies we do not have question subdivisions. If a segment has a question mark, the DA QUESTION is assigned. The second rule assigns DA SOCIAL if the following conditions are fulfilled:

- the segment consists of one token; and
- the segment contains an emoticon or the word *danke* ('thank you').

5. Results

We evaluated three machine learning algorithms trained with different sets of features for each taxonomy. We compared each classifier to the others and conclude which feature set generated the most reliable model. In the numbers reported in this section, we exclude the "0" tag from the evaluation process because it was automatically added and carries no relevant information. This also avoids bias and allows us to more easily compare the evaluation results with DA recognition results on other text types. To achieve a fair comparison, we ran the application with the same feature sets for each method of prediction. We calculated five different measures in order to evaluate the results: precision, recall, f-measure, accuracy and Fleiss' multi-π, which measures the agreement of the predicted DAs with the gold standard annotation. This measure was introduced in order to compare the automatic DA recognition results with human performance. Human inter-annotator agreement can be understood as an estimate of the difficulty of the task and marks an upper limit on the performance to be expected from automatic methods. In all evaluations, we calculated recall and precision for each DA separately and then take the mean, weighted by the prevalence of the DA, to calculate the f-measure.

As we analyzed the gold standard annotations, one DA in each taxonomy has by far the biggest share in occurrences over the corpus: INFORM for the full and the reduced DA taxonomy, and INFORMATION PROVIDING for the minimal taxonomy. As a baseline, we assigned this most frequent DA to all segments. The results are illustrated in Table 2 and are as expected. It is obvious that the minimal DA taxonomy has the best results, since it has the smallest number of DAs in it.

	precision	**recall**	**f-measure**
full	0.052	0.298	0.0892
reduced	0.096	0.403	0.155
minimal	0.231	0.625	0.337

Table 2: Baseline evaluation for each taxonomy.

We used the three feature sets described above, one of them with varying settings, in several combinations:

- **UD**: the user-defined feature set that consists of twelve properties defined in Section 4.
- **L50 / L100**: lexical features for each DA with top 50 or top 100 *tf-idf* values;
- **WE**: global word embeddings.

For all classifiers trained with all possible feature sets we also applied the hand-crafted rules from Section 4. and compared the results. The rules did not improve the predictions much (less than 1% increase of correctly predicted DAs), therefore, we exclude them from further evaluation.

5.1. Evaluation of the Full DA Taxonomy

For all feature sets we trained the models with, the best results are provided by CRFs (see Table 3). The user-defined features alone showed the worst result among all sets. However, the other four sets came up with almost the same results.

A comparison of the results within one classifier shows the best feature selection for Gaussian HMM is the combination of all available sets: ALL (UD+L100+WE). In con-

	MHMM			**GHMM**			**CRF**		
	f	acc.	π	f	acc.	π	f	acc.	π
UD	0.22	0.22	0.33	0.18	0.16	0.25	**0.28**	**0.32**	**0.44**
UD + L50	0.05	0.03	0.42	0.20	0.19	0.49	**0.31**	**0.37**	**0.62**
UD + L100	0.04	0.02	0.42	0.20	0.19	0.50	**0.31**	**0.37**	**0.62**
UD + WE				0.18	0.16	0.45	**0.31**	**0.36**	**0.61**
ALL				0.21	0.22	0.50	**0.31**	**0.37**	**0.62**

Table 3: Dialog act recognition results for the full DA taxonomy (51 DAs).

	MHMM			**GHMM**			**CRF**		
	f	acc.	π	f	acc.	π	f	acc.	π
UD	0.40	0.36	0.60	0.35	0.33	0.60	**0.46**	**0.59**	**0.68**
UD + L50	0.11	0.06	0.44	0.35	0.33	0.60	**0.50**	**0.52**	**0.71**
UD + L100	0.10	0.06	0.44	0.35	0.33	0.60	**0.49**	**0.52**	**0.70**
UD + WE				0.35	0.33	0.60	**0.51**	**0.54**	**0.72**
ALL				0.36	0.34	0.60	**0.51**	**0.53**	**0.71**

Table 4: Dialog act recognition results for the reduced DA taxonomy (12 DAs).

	MHMM			**GHMM**			**CRF**		
	f	acc.	π	f	acc.	π	f	acc.	π
UD	0.63	0.62	0.59	0.51	0.45	0.50	**0.70**	**0.70**	**0.71**
UD + L50	0.42	0.34	0.59	0.50	0.43	0.66	**0.70**	**0.72**	**0.81**
UD + L100	0.43	0.34	0.59	0.50	0.43	0.66	**0.70**	**0.72**	**0.82**
UD + WE				0.49	0.43	0.66	**0.72**	**0.74**	**0.84**
ALL				0.50	0.44	0.66	**0.72**	**0.74**	**0.84**

Table 5: Dialog act recognition results for the minimal DA taxonomy (8 DAs).

trast, for Multinomial HMM it is only the user-defined feature set, while adding extra features dramatically worsens the predictions. This is due to the small training corpus, and lexical features increasing the state space too much for Multinomial HMMs. For CRF and GHMM, combining our user-defined features with the top 50 or top 100 significant words only slightly improves the results compared to only the user-defined features. The word embeddings also yield similar results for CRF.

Although the CRF algorithm gains the best overall results in DA recognition, none of the approaches can recover the majority of dialog acts in the taxonomy. Table 6 shows the number of distinct DAs that were predicted using each method. Almost all DAs that are recognized by all classifiers are from the Information Transfer dimension (ignoring rare dimensions such as Communication Management). Multinomial HMMs without language models found the most DAs (17 of 51). Adding significant words to the user-defined feature set not only decreased the *f-measure* value in MHMM, but also decreased the number of correct recognized DAs (from 17 to 6).

GHMM trained with the combination of the user-defined features extended with word embeddings showed worse results than the CRF trained with the same set (*f-measure* values are 0.18 and 0.31, respectively), but GHMMs found more DAs than CRF (15 and 13, respectively). Moreover, the two algorithms found different DAs. For example, GHMM recognized correctly Request, Suggestion, DSM, but did not find or wrongly predicted TopicIntroduction, Disagreement and GratitudeThank that were found by CRF. This suggests that a combination of learning approaches (for example, through a vote or reranker) might yield improvements in the prediction accuracy even on this small dataset.

	MHMM	**GHMM**	**CRF**
UD	**17**	12	14
UD + L50	6	12	**14**
UD + L100	6	12	**14**
UD + WE		**15**	13
ALL		13	**14**

Table 6: Number of distinct DAs predicted by different classifiers (full DA taxonomy).

5.2. Evaluation of the Reduced DA Taxonomy

Just as in the *Full DA Taxonomy*, CRF produced the best predictions (see Table 4) and the best result within this classifier was reached by the combination of user-defined and *word embedding* features. However, all features combined obtained almost the same results. Extending the user-defined feature set by the top 50 significant words performed more precisely in prediction than extending it by the top 100. This can be explained by the small size of our corpus. By choosing the top 100 significant words, we added almost all words from the corpus, because our gold standard has only 1234 tweets (basically we use a simple bag of words representation without excluding stop words). GHMM showed nearly the same results for all combinations of feature sets and MHMM trained by user-defined features gets significantly better outputs than with the other feature set combinations. MHMM performed the worst, except if the classifier was trained by the user-defined feature set (*f-measure* = 0.40 against 0.35 provided by GHMM).

5.3. Evaluation of the Minimal DA Taxonomy

In the minimal DA taxonomy, CRF reached the best results as well, but different feature sets did not make big differences in the results (see Table 5). Just as in the other taxonomies, MHMM trained only with user-defined features performed better than trained with the other sets. Choosing 100 significant words showed a slight improvement that can be explained by reduction of the taxonomy to eight DAs (excluding the "0" tag). In the minimal DA taxonomy, we could also try to add other language features, because the number of segments assigned to each DA increases, and more detailed language features (for example, bigrams) could improve the results.

GHMM performed better than MHMM, but not if the model was trained with user-defined features. Then the f-measure value for MHMMs was more than 0.10 points larger. In fact, the simple set of 12 user-defined features can already achieve an f-measure of 0.63 in MHMM-based DA recognition, almost 100% improvement over the baseline.

As in the reduced DA taxonomy, the three rarest DAs were not found by any of the classifiers: PCM: 18, OTHER: 12, and OCM: 6. This indicates that the general-purpose dialog act schema (DIT++) used here may not be entirely suitable for classifying tweets, and that other distinctions may be more relevant.

6. Discussion

There are several issues with our approach that we would like to discuss in this section. These issues also open up some clear avenues for improving the results.

Corpus size. Our gold standard corpus consists of 2790 segments (1234 tweets). This is a relatively small corpus for this type of task, especially if it is annotated with the full DA taxonomy that consists of 56 DAs. Not all DAs are present in the gold standard equally: several DAs are not present in the corpus at all, over 20 DAs occur less than five times, and there are only five DAs that occur more than 100 times. By splitting the corpus during 10-fold cross validation, we do not take this information into account. Thus it can happen that DAs occur in the test set, but not in the training set, and as a result the classifier gets no information about these DAs. We attempted to mitigate this problem by merging related DAs from the taxonomy to obtain the reduced and minimal DA sets.

Another issue related to cross-validation is that we have to distribute complete conversations over the 10 folds, not just segments or tweets. Because most conversations are either very short or very long and are randomly chosen for the folds, in some runs we observed that all long conversations are in the test set, rendering the training set smaller than the test set. Consequently, the predictions deteriorated.

Extracting significant words for each DA did not show big improvement in predictions, although in similar works language features play a very important role (Verbree et al., 2006; Zhang et al., 2011). This could again be explained by the small corpus size. Sophisticated previous approaches also employ bigrams or trigrams in DA recognition, which we cannot do for the same reason.

Annotation. Since we asked novice annotators to annotate Twitter conversations with the full DA Taxonomy (56 DAs), the inter-annotation agreement is relatively low: Fleiss' multi-π = 0.56 (in comparison to the work of Stolcke et al. (Stolcke et al., 2000) that achieved very good agreement of Cohen's κ = 0.8 with 42 DAs). In future studies, we could divide the annotation procedure into two separate tasks: one for the segmentation and a second one for annotating the segments. Another option is crowdsourcing the annotations. We have opted against crowdsourcing for two main reasons. First, it is harder to find naive annotators for languages other than English. Second, fine-grained DA annotation requires a familiarity with the taxonomy that cannot be achieved without training. Our student annotators, in addition to genuine disagreements about annotations, also made many errors that had to be corrected during costly reanalysis and curation.

In order to get better agreement, the annotators should be better trained (for example, how to distinguish question types). Another option for untrained or minimally trained annotators is to simplify the DA taxonomy or to binarize annotation decisions and present them one at a time. Further, the annotation guidelines should be more elaborated. For example, there should be clear instructions on how to segment a tweet:

1. exclude user mentioning in the beginning of the tweet by assigning "0" DA;
2. include punctuation in the previous segment;
3. include links to the segment if they are semantically connected to the segment;
4. treatment of emoticons, etc.

Choosing the DA for the first segment, the dialog opening, can also be a cause for confusion. In the full DA taxonomy, we have two DAs that can describe a conversation start: TOPICINTRODUCTION and OPEN. A segment can fit both DAs, since opening a conversation often happens with introducing a topic. In addition, some annotators used INFORM. Since we required that each segment can only be assigned one DA, this led to regular disagreement. In our current annotation efforts, we therefore allow segments to be assigned several DAs if they fulfill both functions. This is in line with previous work by (Bunt et al., 2010) and (Core and Allen, 1997).

Adequacy of the schema. Also, during the annotation process problems were caused by the DA INFORM and its branches due to their diffuse boundaries. As always in DA annotation projects, most segments are assigned a variant of INFORM. However, these do not all correspond to identical communicative functions. In the group of segments annotated with the DA INFORM, we can find varying dialog functions like factual statements, meta-commentary, discourse management, opinions, and also sarcastic/ironic statements (Zarisheva and Scheffler, 2015). When annotators felt confused or unsure, they often assigned the superordinated DA INFORMATION PROVIDING FUNCTIONS to the segment. For example, some annotators were very reluctant to label sarcastic comments with INFORM. On social media like Twitter, we find many different types of dialog. Not all can be captured well with a dialog act taxonomy that is mainly based on informational exchanges (between humans or humans and machines). It seems necessary to revise the DA taxonomy to better reflect the different types of statements common on Twitter. This should not only lead to more accurate annotations, but also improve downstream applications that need to distinguish between factual statements and opinions, straight talk and sarcasm, etc.

Features and models. Finally, in order to gain better results we need to not only enlarge the corpus but also to train models with additional linguistic features (bigrams, trigrams, part-of-speech, first and last words of a segment, etc.). In further work, other classifiers can be implemented. Related works show good performance of Decision Trees and Support Vector Machine extended with Hidden Markov Models (HMM-SVM).

7. Conclusion

In this paper, we introduced our approach to supervised dialog act classification for German Twitter conversations. In contrast to previous work, we viewed entire conversations as dialogs and classified individual segments within tweets (a tweet can contain more than one segment). In addition, we used fine-grained dialog act annotations with a taxonomy containing 56 categories.

To automatically assign DAs to segments, we used three supervised learning algorithms (HMM with Multinomial and Gaussian distributions, and CRF) and trained them with different sets of features that we extracted from the training data. We established twelve user-defined features indicating structural and lexical properties of the dialog. In addition, we used the top 50 or 100 tf-idf-ranked unigrams for each dialog act as a feature set for the segment to be classified. Alternatively, we incorporated word embeddings to train two of the classifiers (GHMM and CRF). The two language models (tf-idf and word embeddings) yielded similar results for our experiments. However, larger datasets might lead to an advantage for word embeddings, where the feature space remains small without losing too much information.

We trained the three classifiers with various combinations of the chosen features. For all taxonomies, the CRF classifier with the full feature set performed the best. For the full DA taxonomy, we achieve an f-measure of up to 0.31, for the reduced taxonomy, up to 0.51, and minimal taxonomy, 0.72.

The results show that dialog act recognition on Twitter conversations is quite reliable for small taxonomies. The minimal taxonomy of 8 DAs can be compared to previous work on speech act classification in social media posts. Our results are better than previously reported results for speech act classification on entire tweets by (Zhang et al., 2011), who report F1=0.695 for 5 classes. The improvement could be due to two factors: (i) Our classifiers explicitly model the sequential structure of conversations, whereas (Zhang et al., 2011) use SVM classifiers on individual tweets without context features. (ii) We segment tweets into utterances first (segmentation is not part of this work), while (Zhang et al., 2011) assign exactly one speech act to a tweet. This may lead to classification problems in the many cases where one tweet is composed of several distinct speech acts (over 30% in our corpus).

Similarly, our results improve over (Arguello and Shaffer, 2015)'s work on classifying MOOC forum posts into seven speech acts (average precision of their best feature set, around 0.65; ours, 0.70). Although (Arguello and Shaffer, 2015) take dialog structure into account by incorporating some history features, explicit sequence labelling approaches such as CRF may be better able to capture the interdependencies in dialog.

It would be interesting to study which factors still limit the performance of dialog act classification on social media text (such as this work) compared to previous work on human face-to-face dialog. In addition to the amount of available data and the annotation quality, other disadvantages may be the unavailability of prosodic features, non-standardized spelling and vocabulary, and additional types of interaction or dialog acts (such as the prevalence of sarcasm on Twitter). Our work compares well to some previous DA recognition projects such as (Ang et al., 2005) on multiparty meetings, but stays far behind large efforts like (Stolcke et al., 2000), who report recognition accuracy of 0.71 on the Switchboard corpus with 42 DAs (our work: 0.37 for 51 DAs).

To achieve such competitive results, we need not only much more data, but should also revise the annotation schema to better reflect social media content. In the absence of additional annotation efforts, cross-domain transfer may also be used to improve DA prediction results on this new domain. In addition, different approaches such as combined segmentation and DA recognition and the combination of existing DA classifiers through voting or reranking, should be explored.

8. Acknowledgements

The authors are grateful to the students of the 2014/15 University of Potsdam seminar "Dialogs in Twitter" for annotating the data. We thank the anonymous reviewers for their helpful comments.

9. Bibliographical References

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the 17th CONLL*, pages 183–192, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ang, J., Liu, Y., and Shriberg, E. (2005). Automatic dialog act segmentation and classification in multiparty meetings. In *ICASSP*, pages 1061–1064.

Arguello, J. and Shaffer, K. (2015). Predicting speech acts in MOOC forum posts. In *Proceedings of the 9th International AAAI Conference on Web and Social Media*.

Artstein, R. and Poesio, M. (2008). Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Bunt, H., Alexandersson, J., Carletta, J., Choe, J.-W., Fang, A. C., Hasida, K., Lee, K., Petukhova, V., Popescu-Belis, A., Romary, L., et al. (2010). Towards an ISO standard for dialogue act annotation. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Core, M. and Allen, J. (1997). Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, pages 28–35. Boston, MA.

Forsyth, E. N. and Martell, C. H. (2007). Lexical and discourse analysis of online chat dialog. pages 19–26.

Honeycutt, C. and Herring, S. C. (2009). Beyond microblogging: Conversation and collaboration via twitter. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–10. IEEE.

Ritter, A., Cherry, C., and Dolan, B. (2010). Unsupervised modeling of Twitter conversations. In *Proceedings of NAACL*.

Scheffler, T. (2014). A German Twitter snapshot. In N. Calzolari et al., editor, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Verbree, D., Rienks, R., and Heylen, D. (2006). Dialogue-act tagging using smart feature selection: Results on multiple corpora. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 70–73. IEEE.

Zarisheva, E. and Scheffler, T. (2015). Dialog act annotation for Twitter conversations. In *Proceedings of SIGDial16*, pages 114–123, Prague, Czech Republic, September. Association for Computational Linguistics.

Zhang, R., Gao, D., and Li, W. (2011). What are tweeters doing: Recognizing speech acts in twitter. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Appendix: Full Dialog Act Schema

```
ADF Action Discussion Functions
  Commissive
    Offer
      Promise
      Threat
      AddressRequestSuggestion
        Accept
        Decline
  Directive
    Request
    Suggestion
    AddressOffer
      Accept
      Decline
IT Information Transfer Functions
  IP Information Providing Functions
    Inform
      Answer
      Agreement
      Disagreement
        Correction
  IS Information Seeking Functions
    Question
      ChoiceQuestion
      SetQuestion
      PropQuestion
        CheckQuestion
DSM Discourse Structure Management
  Open
  TopicIntroduction
  TopicShift
OCM Own Communication Management
  Error (Error signaling)
  Retraction
  SelfCorrection
PCM Partner Communication Management
  Completion
  Correct (Correct-misspeaking)
SOCIAL Social Obligations Management
  Apologize
    Apology
    ApologyDownplay
  Bye
    InitialBye
    ReturnBye
  Gratitude
    Thank
    ThankDownplay
  Introduce
    InitialIntroduce
    ReturnIntroduce
  Salutation
    Greet
    ReturnGreet
OTHER
```

Normalisation of Lithuanian Social Media Texts: Towards Morphological Analysis of User-Generated Comments

Andrius Utka, Darius Amilevičius
Vytautas Magnus University
K. Donelaičio 52-206, Kaunas, Lithuania
E-mail: a.utka@hmf.vdu.lt, d.amilevičius@if.vdu.lt

Abstract

In this paper, we present a preliminary research on the normalisation of Lithuanian social media texts. Specifically, the paper deals with language normalisation issues in Lithuanian user-generated comments in the three popular websites: *Lietuvos Rytas* (Lithuanian Morning)*, Verslo žinios* (Business News), and *Delfi.lt*. We have established the proportion of out-of-vocabulary (OOV) words in the dataset by using a standard Lithuanian tokenizer and a morphological analyser from the newly developed Information System for Semantic and Syntactic Analysis of the Lithuanian Language (LKKSAIS). A detailed qualitative analysis of extracted OOV words is presented, where specific aspects of Lithuanian social media texts are determined: namely, the extent of missing diacritics, as well as other prevalent error types. A standard Lithuanian spell checker is used for the restoration of missing diacritics and correction of other errors in user-generated comments, which considerably improves the morphological analysis.

Keywords: out-of-vocabulary words, user-generated comments, morphological analysis, Lithuanian

1. Introduction

Since the start of this century, normalization of social media texts is in the focus of the NLP community. The increased interest in this topic is based on the two main factors: firstly, if properly analysed social media texts offer many possibilities for exploration in marketing and security sectors; secondly, non-standard language of social media texts hides interesting information, which needs to be deciphered, i.e. normalized.

A number of different approaches are used to normalize non-standard language texts. In summary, two major trends can be distinguished: dictionary-based approaches and statistical machine translation (SMT) approaches. While the dictionary-based approaches are dependent on the size of dictionaries and methods of identifying in-vocabulary (IV) and out-of-vocabulary words (OOV), the SMT approaches are dependent on large quantities of parallel training data.

It should also be mentioned that most of research and methodology is directed towards English social media, especially tweets on Twitter, SMS messages, blogs, etc., while other languages are not so well covered. This is due to the fact that English, being a lingua franca of the world, dominates all types of social media. Besides, there is considerably better availability of English social media corpora and NLP tools, which stimulates further research.

Baldwin et al. (2015) have shown that there are a dozen of languages besides English which are strongly represented in different social media, e.g., Japanese, Portuguese, Spanish, German, Russian, French, Indonesian, Dutch, Malay, Italian, and Chinese. Likewise, the research of social media texts of for these languages are rather well covered.

However, the situation of smaller languages, spoken by less than 5m speakers, is rather different. Some of them are official state languages (e.g. Estonian, Georgian, Latvian, Lithuanian, Maltese, Norwegian, Slovene, and others), while others are regional languages (e.g. Galician in Spain, Sicilian in Italy, languages in Indonesia, India, Iran, and many others[1]). Typically, NLP instrumentation of such languages is rather humble, these languages have much less explicit presence in top international social media platforms, and the available language resources are also limited. Nevertheless, many speakers of these smaller languages use social media channels for communication, which are not necessarily among the top international social media platforms.

This paper will present a case of the Lithuanian language, which is currently spoken by approximately 4m people in the world. The most popular social networks in Lithuania are: *YouTube* (77%), *Facebook* (68%), *Google+* (27%), *One.lt* (21%), *Draugas.lt* (15%), *Twitter* (7%)[2], LinkedIn (6%), and others.

While usage of social media networks by Lithuanians is similar to the usage by other Europeans, there is one feature of internet usage that distinguishes Lithuanians from other Europeans. According to DESI 2015 and 2016 reports, Lithuania ranks first in the EU, according to the percentage of individuals (94%) who used Internet for reading online news in the last 3 months (while EU average is 68%.)[3]. Consequently, comment sections of news articles are very important type of social media communication for Lithuanians.

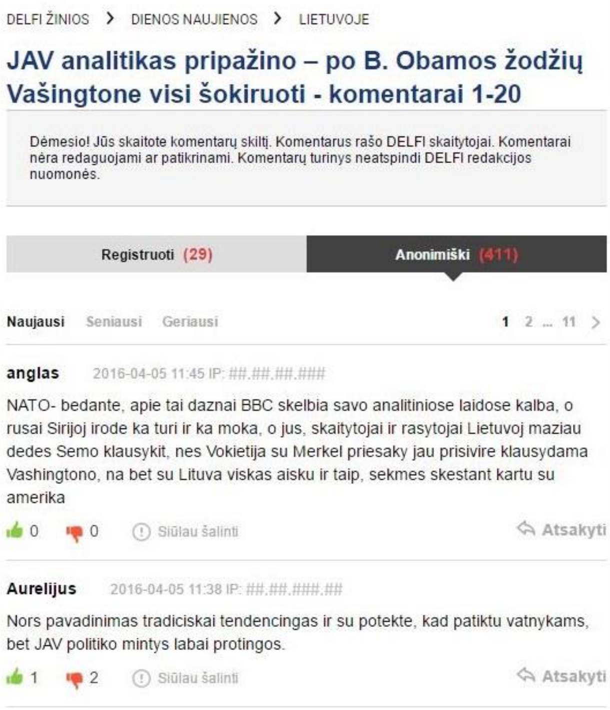
The comment sections are attached to news articles and readers can anonymously publish their comments that can be seen by all other readers. Very often these comments turn into public forums or chats on the topic of the article.

[1] Source: *https://www.ethnologue.com/*

[2] Lithuanians mostly tweet in English.

[3] https://ec.europa.eu/digital-single-market/en/scoreboard/lithuania

Figure 1 illustrates the visual layout of the comment section in the most popular Lithuanian news portal *DELFI.LT*[4].


DELFI ŽINIOS › DIENOS NAUJIENOS › LIETUVOJE

JAV analitikas pripažino – po B. Obamos žodžių Vašingtone visi šokiruoti - komentarai 1-20

Dėmesio! Jūs skaitote komentarų skiltį. Komentarus rašo DELFI skaitytojai. Komentarai nėra redaguojami ar patikrinami. Komentarų turinys neatspindi DELFI redakcijos nuomonės.

anglas 2016-04-05 11:45 IP: ##.##.##.###

NATO- bedante, apie tai daznai BBC skelbia savo analitiniose laidose kalba, o rusai Sirijoj irode ka turi ir ka moka, o jus, skaitytojai ir rasytojai Lietuvoj maziau dedes Semo klausykit, nes Vokietija su Merkel priesaky jau prisivire klausydama Vashingtono, na bet su Lituva viskas aisku ir taip, sekmes skestant kartu su amerika

Aurelijus 2016-04-05 11:38 IP: ##.##.###.##

Nors pavadinimas tradiciskai tendencingas ir su potekte, kad patiktu vatnykams, bet JAV politiko mintys labai protingos.

Figure 1: Comments in DELFI.LT news website.

This form of transmission of personal opinions is thriving in Lithuanian media and is not hindered by the facts that news websites reveal IP addresses of each commentator and that several homophobic or bomb-threat comments have ended in the public revelation of commentators' identities or even in courts. For these reasons, we have selected user-generated comments as the object of research.

In Lithuania, the situation of NLP tools and language resources somewhat improved after the completion of the large scale project SEMANTIKA (Vitkutė-Adžgauskienė et al., 2016; Vileiniškis et al., 2015). During the project, a number of basic NLP tools have been developed and the Information System for Semantic-Syntactic Analysis of the Lithuanian language has been built (LKSSAIS), which includes the NLP pipeline for the analysis of Lithuanian language texts. In the current experiment, only three basic tools from the infrastructure have been used.

As this paper is the first attempt to focus on the normalisation problem of Lithuanian media texts, initially, we decided to assess the extent of noise in user-generated comment sections of the three most popular Lithuanian news websites. As we lack the reference corpus for Lithuanian social media texts in order to assess precision levels of used tools, our analysis is limited to out-of-vocabulary words (OOV). The proportion of OOV words is a good estimator of noise levels in social media texts, as they have a major influence on the performance of POS taggers (e.g. see Giesbrecht & Evert, 2009; Neunerdt, 2013). Thus, firstly, we establish the proportion of out-of-vocabulary words (OOV) in social media texts while using a standard tokenizer and a morphological analyser for the Lithuanian language. Secondly, as the majority of OOV words occur due to the missing diacritics, we will manually classify the list of OOV words in order to establish the extent of OOV words that are written without diacritics, as well as other problems, and, finally, we will test a standard Lithuanian language spell checker for dealing with OOV words.

2. Related Work

Previously, the Lithuanian social media language has received some attention from the Lithuanian language researchers. We should mention Rykliené's work (2000, 2001), who aptly showed how the spontaneously written Lithuanian language is similar to speech, as well as papers by Marcinkevičienė (2006), Miliūnaitė (2008), and Žalkauskaitė (2011). All these studies reveal the functions and diversity of the spontaneous written Lithuanian, however, they are based on small datasets and detailed manual analyses. Although some useful observations and explanations are provided, they have limited practical applicability to effective processing of new large datasets.

Only recently, a group of Lithuanian researchers has analysed Lithuanian social media texts from the perspective of computational linguistics (Kapočiūtė-Dzikienė et al., 2013, 2015). The paper by Kapočiūtė-Dzikienė et al., (2013) deals with the problem of normalisation in relation to the classification of sentiments in internet comments. The paper reports that the proportion of out-of-vocabulary words is 25%, when a morphological analyser *Lemuoklis* (Zinkevičius, 2000) is used. It is notable that the problem of missing diacritics is solved in a rather unusual approach: diacritical letters are simply converted to non-diacritical ones, both in dictionaries and in analyzed texts, thus decreasing the number of OOV words in their dictionary-based approach for sentiment detection.

Papers that deal with the estimation of noise in social media texts of other languages are also relevant to this research. We should definitely mention the work by Baldwin et al. (2013), Gadde et al. (2011) on English, Rello & Baeza-Yates (2012) on English and Spanish (although their estimations are based on different methodology), as well as papers by Giesbrecht & Evert (2009) and Neunerdt et al., (2013a and 2013b) on the evaluation of POS taggers, when used on German social media texts.

3. The Lithuanian Language

The Lithuanian language belongs to the branch of the Baltic languages. Lithuanian has a relatively free word order and a very rich morphemic structure: nouns, adjectives and pronouns have 7 cases in singular and 7 in plural, as well as 3 additional locative cases used in certain contexts. While nouns can have two gender forms (masculine and

[4] According to a recent survey *Delfi.lt* ranks first among Lithuanian portals, *Lithuanian morning* – 4th, and *Business News* – 6th (source: http://audience.gemius.com.

feminine), adjectives, numerals, pronouns and participles can have three gender forms (masculine, feminine, and neuter). The verbal system of Lithuanian is especially rich as beside the regular verbal tenses, moods, and aspects, it has 13 types of participles.

The Lithuanian alphabet is a Latin alphabet, consisting of 32 letters. There are 7 letters (*a, e, i, u, c, s, z*) which may be marked by diacritics to designate different sounds, inflections of different cases, and writing conventions. As "e" and "u" may be marked by two different diacritics, there are 9 different letters with diacritics in the Lithuanian alphabet:

Vowels (6): *Ą, ą; Ė, ė; Ę, ę; Į, į; Ų, ų; Ū, ū*
Consonants (3): *Č, č, Š, š, Ž, ž*

The diacritical letters make up 6.9% of all letter usage in Lithuanian texts (Grigas & Juškevičienė, 2015), and 39.1%[5] of Lithuanian word forms contain at least one diacritical letter. These facts are important to the normalisation of Lithuanian media texts, as many Lithuanians, similarly to the speakers of other diacritical languages, when writing in various social media platforms, tend to avoid diacritical letters replacing them with non-diacritical ones. Wordforms without correct diacritics are understandable by human-readers, but they create additional problems to POS tagger, then to parser, and then to all other tools and services that are dependent on correct POS tags: e.g. *šeima* ('family') becomes *seima* (OOV), *karšto* ('hot') becomes *karsto ('*coffin*')*, *lovą* ('bed' in Accusative case) – lova (Nominative case), etc.

4. Dataset

For this research, we have compiled a dataset[6] which is made from comment sections in three popular Lithuanian news websites, namely *DELFI*[7]*, Verslo žinios*[8] *(Business News),* and *Lietuvos Rytas*[9] *(Lithuanian Morning).*

For comparison reasons, we have also compiled a 50 thousand word corpus from news' articles written in the standard Lithuanian. The summary statistics is given in Table 1:

News Website	Average length	Texts	Number of Word forms
DELFI	29.55	1,724	50,951
Business News	41.73	416	17,359
Lithuanian Morning	31.95	1,344	42,937
Average/Total:	*31.71*	*3,484*	*110,485*
News Corpus			50,078

Table 1: Summary Statistics for the dataset.

5. Tools

As mentioned, only three components from LKSSAIS infrastructure have been used for the analysis of the social media texts. The components used are a tokenizer (*lex)*, a morphology tagger (*morphology*), and a spell checker (*spelling*). All three components have been developed to deal with the Lithuanian web news texts that are written in standard Lithuanian.

Both the morphological tagger and the spell checker are based on the *Hunspell*[10] engine. Currently, its morphological database contains more than 112,000 Lithuanian lemmas and achieves the precision of 95% and recall of 95% on web news texts (Vitkutė-Adžgauskienė et al., 2016). The morphological tagger generates annotations that contain (1) lemmas; (2) a full set of rich morphological information which is rendered by MULTEXT-East tag-set[11], and (3) stems in *json* format. For example[12]:

```
‟Lietuvoj litras Borjomi 1.49 litras
dyzelino 0.80 Nereikia ir emiratu″
(In Lithuania a litre Borjomi 1.49 a litre
dieseline 0.80 You don’t need Emirates)

{"msd":[[["Lietuva","Npfslng"]],
[["litras","Ncmsnn-"]],
[["Borjomi","X-"]],
[["1","M----d-"]],
[[".","Tp"]],
[["49","M----d-"]],
[["litras","Ncmsnn-"]],
[["dyzelinas","Ncmsgn-"]],
[["0","M----d-"]],
[[".","Tp"]]
[["80","M----d-"]],
[["nereikėti","Vgmp3---y--ni-"],
["nereikėti","Vgmp3s--y--ni-"],
["nereikti","Vgmp3s--y--ni-"],
["nereikėti","Vgmp3p--y--ni-"],
["nereikti","Vgmp3p--y--ni-"],
["nereikti","Vgmp3---y--ni-"]],
[["ir","Qg"],["ir","Cg"]],
[["emiratu","X-"]]]

"stem":["Lietuv","litr","Borjom","1",".","4
9","litr","dyzelin","0",".","80","Nereik","
ir","emirat"}
```

For ambiguous word forms, the tagger produces a list of alternatives with the most probable one at the top (e.g., *nereikėti,* En. *"You don't need"*). The current implementation of the tagger does not tag OOV words. The OOV words are marked by *X-* or *X* tags.

For our experiment, we have used two pipelines from the

[5] Based on the 1m word frequency dictionary (Utka 2009).

[6] The corpus will be deposited and shared via CLARIN infrastructure, as soon as Lithuanian CLARIN-LT repository is set up.

[7] http://www.delfi.lt

[8] http://vz.lt

[9] http://www.lrytas.lt

[10] http://hunspell.github.io/

[11] http://nl.ijs.si/ME/V4/msd/html/index.html

[12] Line breaks are added to ensure readability.

above mentioned infrastructure: 1) a regular one (*lex -> morphology*), and 2) a language normalization one (*lex -> spelling -> morphology*).

6. Out-of-Vocabulary Words

6.1 The Proportion of OOV Words

The proportion of OOV was established by using a standard tokenizer *lex* and a morphological analyser *morphology* and by extracting all occurrences of unrecognized words which are marked by X or X- in annotated texts.

News Website	OOV	%
DELFI	8,721	17.116
Business News	2,972	17.121
Lithuanian Morning	8,032	19.044
Total/Average:	*17,825*	*17.853*
News Corpus	1,757	3.509

Table 2: OOV words in user-generated comments and in news corpus.

Table 2 shows that comments from *Lithuanian Morning* contain by 2% more OOV words than *Delfi* and *Business News*. Unsurprisingly, the news corpus has by far the cleanest language of all. The proportion of OOV words is by 7 percentage points lower than the one reported in the experiment by Kapočiūtė et al. (2013), where they report ~25% of OOV words in a similar dataset, however, with a different tagger *Lemuoklis*.

How does the OOV proportion between 17.12-19.04% in the Lithuanian data relate to other languages and genres? In order to have similar conditions for comparison, we will mention only the results that have been generated by standard tools and which have not been trained on the web data.

For the English language, Baldwin (Timothy) et al. (2013) report that two Twitter datasets have 24.0% and 24.6% of OOV words, comments − 19.8%, forums – 18.1%, blogs 20.6%, while using GNU aspell dictionary v0.60.6.1. The numbers are well comparable to the Lithuanian results, however, high proportion of OOV words in Wikipedia (19.0%) and BNC (16.9%) come as a surprise, as these sources are supposed to be written in close to standard English.

Gade et al. (2011) report that the proportion of OOV words in English SMS messages is 34.2% while using an English POS tagger which is trained on the Wall-Street Journal (WSJ) corpus. This is a confirmation that SMS messages are by far the noisiest social media genre of all.

With respect to the German language, Neunerdt et al. (2013) report that the proportion of OOV words in web comments corpus is 14.71% when trained on standard German TIGER corpus. The number is by 3-5% lower than the Lithuanian or English data.

Unsurprisingly, the proportion of OOV words is very different and hardly comparable as the numbers are based on different languages, genres, taggers and selection methods.

6.2 Qualitative Analysis of OOV Words

Classification of OOV words according to error type is the next step in our analysis. Although we already know that the major cause of OOV words in Lithuanian user-generated comments is the lack of diacritics, the extent of the problem is not clear. Besides we would like to establish the extent of other major errors. This classification could help us to select a strategy for normalisation.

The classification was performed manually by assigning each OOV word form in the extracted list of OOV words to appropriate categories. We marked each word form for missing diacritics and for other error categories. We did not restrict the classification of errors to any existing schemes used for other languages, but rather created our own fine-grained classification of 21 categories (see Table 4). Some word forms have been assigned to more than one error type, e.g. *gyvenma* should be spelled *gyvenimą*, thus, in this case, the diacritic is missing and the word is also misspelt.

As such classification is a time consuming activity, we have classified the sample of most frequent 5,500 OOV word forms from the DELFI comments.

OOV	Correct	Freq.	Diacritic
1. i (prep. ‘to’)	į	188	d
2. is (prep. ‘from’)	iš	172	d
3. cia (prep. ‘here’)	čia	126	d
4. uz (prep. ‘behind’)	už	106	d
5. ka (part. ‘what’)	ką	105	d
6. ju (pron. ‘their’)	jų	67	d
7. del (prep. ‘for’)	dėl	66	d
8. as (pron. ‘I’)	aš	66	d
9. zmones (n. ‘people’)	žmonės	61	d
10. musu (pron. ‘ours’)	mūsų	57	d
…	…	…	…
1840. seimyna (n. ‘family’)	šeimyna	1	d
Total:		5,500	

Table 3: 10 most frequent OOV words in the testing sample.

Table 3 presents ten most frequent OOV words. It is evident that they are dominated by prepositions and pronouns. These 10 words cover 18 per cent of our testing sample.

A more general summary of error types is given in Tables 4 and 5. 74.6% of OOV words are without diacritics, while most prominent errors among remaining 26.4% of OOV words are named entities (5.9%), incorrect word boundaries (3.9%), foreign words (2.7%), and misspellings (2.4%).

Missing diacritics	4105	74.6%
Other errors	1395	25,4%
Total OOV words	*5500*	*100,0%*

Table 4: Proportion of OOV words that lack diacritics.

The analysis suggests that a diacritic restoration or a simple spell check may considerably reduce the number of OOV words. Besides, we found out that most of the named entity problems are related to the taggers’ case-insensitive treatment of proper nouns, thus among unrecognized forms are *rusija* ('Russia'), *lietuvoje* ('Lithuania', in Locative case), *lietuva* ('Lithuania', in Nominative case), and other common proper names.

Error Type	Freq.	%
Named entities	**326**	**5.9**
Incorrect word boundaries	**213**	**3.9**
Foreign words	**149**	**2.7**
Misspellings	**130**	**2.4**
Slang	98	1.8
Acronyms	88	1.6
Abbreviations	86	1.6
New original word forms	83	1.5
Creative writing	60	1.1
Unrecognized correct words	33	0.6
Swear words	30	0.5
Numbers and dates	29	0.5
Intentional Incorrect word boundaries	25	0.5
Symbols	25	0.5
Hyphenation	21	0.4
Multiple letters	16	0.3
Non-Lithuanian alphabet	16	0.3
Old words	4	0.1
Not found	4	0.1
Fixed expressions	3	0.1
Dialect words	2	0.0

Table 5: Error types of OVV words.

7. Automatic Normalization by a Spell Checker

As the last step in our experiment, we tested the effectiveness of the most obvious tool for normalisation, a spell checker.

We have used a standard Lithuanian spell checker (*spelling*) from LKSSAIS infrastructure before morphological analysis, thus using the pipeline: tokenizator + spell checker + morphological analyser. The analysed dataset has not been pre-processed in any way.

As the spell checker is using the same database as morphological analyser, it identifies the same list of OOV words and attempts to correct them.

We consider a correction by spell checker as "correct", when the first spell checker's suggestion is correct, and "incorrect", when spell checker is suggesting a wrong word form as the first suggestion (although other alternatives may be correct). "Not-corrected" are the cases when spell checker does not provide any corrections, thus an OOV word remains unchanged. Table 6 presents the results of this experiment.

Correction Type	%
Correct	54,2
Incorrect	27,1
Not-corrected	18,7

Table 6: Types of corrections by a spell checker.

Table 6 shows that the usage of the Lithuanian standard spell checker would shorten the list of OOV words by half, but it would also replace 25.9% of OOV words by wrong word forms, and 19.0% of OOV words would remain uncorrected.

The analysis of test data has also shown that sometimes word forms without diacritics coincide with legitimate word forms and, therefore, they cannot be detected by the spell checker as OOV words. Such errors are especially difficult to tackle and will require a special adaptation of the spell checker.

8. Conclusions

The paper presented a preliminary research of normalisation of Lithuanian social media texts. Specifically, we focused on out-of-vocabulary words in Lithuanian user-generated comments in three popular websites: *Lietuvos Rytas* ('Lithuanian Morning'), *Verslo Žinios* (Business News) and *Delfi.lt.*

While using a standard Lithuanian tokenizer and a morphological tagger, we have extracted a list of OOV words. The proportion of OOV words over the whole corpus is 17.853%. The qualitative analysis of OOV words showed that 74.6% of OOV words are due to the lack of diacritics, while other prevalent causes are: named entities (5.9%), incorrect word boundaries (3.9%), foreign words (2,7%), and misspellings (2.4%).

We have shown that a standard Lithuanian spell checker, if used before the POS tagger, may considerably improve results of morphological analysis without any pre-processing. This approach is a cost-effective alternative to more expensive language normalisation systems and can be used as a baseline for normalisation of Lithuanian social media texts.

For the future research, we plan to extend our analysis to other genres of Lithuanian social media texts and to investigate different diacritic restoration options. We also plan to compile a golden standard corpus for testing and training tools for the analysis of Lithuanian social media texts. More precise analysis will open new opportunities for more advanced analysis: commentary ranking by relevance, sentiment analysis, parsing, etc.

9. Acknowledgments

The research was funded by CLARIN-LT consortium.

In the paper, we used tools from LKSSAIS infrastructure, which was funded by the European Regional Development Fund.

We would like to thank Dr. Jurgita Vaičenonienė for valuable comments and corrections of the paper.

10. Bibliographical References

Baldwin, T., Cook, P., Lui, M., Mackinlay, A. and Wang, L. (2013). How Noisy Social Media Text, How Diffrnt Social Media Sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan*, pp 356--364.

Gadde, P., Subramaniam, L. V. and Faruquie, T. A. (2011). Adapting a WSJ Trained Part-of-Speech Tagger to Noisy Text: Preliminary Results. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for*

Noisy Unstructured Text Data, pp. 5:1--5:8.

Giesbrecht, E., Stefan, E. (2009). Is Part-Of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus. In *Proceedings of the 5th Web as Corpus Workshop (WAC5,* Donostia (2009).

Grigas, G., Juškevičienė, A. (2015). Raidžių dažnių lietuvių ir kitose kalbose, vartojančiose lotyniškus rašmenis, analizė (Letter Frequency Analysis of Lithuanian and Other Languages Using the Latin Alphabet). *Santalka. Filologija, Edukologija 23*, pp. 81--91.

Kapočiūtė-Dzikienė, J., Krupavičius, A. and Krilavičius, T., (2013). A Comparison of Approaches for Sentiment Classification on Lithuanian Internet Comments. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing. Workshop, Sofia, Bulgaria*, pp. 2--11.

Kapočiūtė-Dzikienė, J., Utka, A. and Šarkutė, L. (2015). Authorship Attribution of Internet Comments with Thousand Candidate Authors. In *Proceedings of ICIST 2015: 21st International Conference on Information and Software Technologies*. Springer International Publishing, pp. 433--448.

Neunerdt, M., Reyer, M. and Mathar, R. (2013a). Part-of-Speech Tagging for Social Media Texts. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology*.

Neunerdt, M., Reyer, M. and Mathar, R. (2013b). A POS Tagger for Social Media Texts Trained on Web Comments. *Polibits 48,* pp. 61--68.

Rello, L., Baeza-Yates, R. (2012). Social Media Is NOT that Bad! The Lexical Quality of Social Media. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM 2012), Dublin, Ireland.*

Ryklienė, A. (2001). Elektroninis diskursas: kalbos ypatybės ir stilius. (Electronic Communication: Talking while Writing). Doctor dissertation. Kaunas: VDU leidykla.

Ryklienė, A. (2000). Bendravimas internetu: kalbėjimas rašant (Electronic Communication: Talking while Writing). *Darbai ir dienos 24,* pp. 99--107.

Utka, A. (2009). *Dažninis rašytinės lietuvių kalbos žodynas.* (Frequency dictionary of the Lithuanian language). Kaunas: VDU leidykla.

Vileiniškis, T., Šukys, A. and Butkienė, R. (2015). An Approach for Semantic Search over Lithuanian News Website Corpus. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR 2015), Lisbon, Portugal,* pp. 57--66.

Vitkutė-Adžgauskienė, D., Utka, A., Amilevičius, D. and Krilavičius, T. (2016). NLP Infrastructure for the Lithuanian Language. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'14)*. Portorož, Slovenia: ELRA.

Zinkevičius, V. (2000). Lemuoklis – morfologinei analizei. (Morphological Analyser *Lemuoklis*). *Darbai ir dienos 24*, 245--273.