Noise or music? Investigating the usefulness of normalisation for robust sentiment analysis on social media data

Cynthia Van Hee — Marjan Van de Kauter — Orphée De Clercq — Els Lefever — Bart Desmet — Véronique Hoste

Language and Translation Technology Team, Dep. of Translation, Interpreting and Communication, Ghent University. Groot-Brittanniëlaan 45, 9000 Ghent, Belgium

ABSTRACT. *In the past decade, sentiment analysis research has thrived, especially on social media. While this data genre is suitable to extract opinions and sentiment, it is known to be noisy. Complex normalisation methods have been developed to transform noisy text into its standard form, but their effect on tasks like sentiment analysis remains underinvestigated. Sentiment analysis approaches mostly include spell checking or rule-based normalisation as preprocessing and rarely investigate its impact on the task performance. We present an optimised sentiment classifier and investigate to what extent its performance can be enhanced by integrating SMT-based normalisation as preprocessing. Experiments on a test set comprising a variety of user-generated content genres revealed that normalisation improves sentiment classification performance on tweets and blog posts, showing the model's ability to generalise to other data genres.*

RÉSUMÉ. *Ces dernières années ont été marquées par un intérêt croissant pour l'analyse des sentiments sur les réseaux sociaux. Bien que ces derniers soient une source de données utile, ils sont connus pour être bruité. Diverses méthodes de normalisation complexe existent pour transformer du texte bruité en sa forme standard. Cependant, la plupart des études à l'analyse des sentiments incluent la normalisation basée sur des règles et ne recherchent guère sa valeur ajoutée à la classification. Nous présentons un système pour l'analyse des sentiments optimisé et examinons si sa performance s'améliore si la normalisation basée sur la traduction automatique statistique est intégrée. Des expériences avec un corpus de test varié démontrent une meilleure performance si la normalisation est incluse, non seulement pour le genre Twitter, mais aussi pour le genre blog, ce qui démontre la capacité de généralisation du modèle.*

KEYWORDS: *sentiment analysis, social media, optimisation, normalisation.*

MOTS-CLÉS : *analyse de sentiments, réseaux sociaux, optimisation, normalisation.*

TAL. Volume 58 – n°1/2017, pages 63 à 87

1. Introduction

Emotions and sentiment play a key role in successful communication and relationships, and the opinions and beliefs of others play a large part in how we evaluate the world or make decisions. The study of sentiment and its related concepts such as opinions and attitudes is known as automatic *sentiment analysis* or *opinion mining*. With the growing amount of opinionated data on the web, sentiment analysis applications have found their way into various research fields and companies showing interest in understanding how consumers evaluate their goods and services. Although the first studies on automatic sentiment analysis appeared in the early 2000's when researchers focused on newswire text (Wiebe, 2000; Pang *et al.*, 2002), the thriving interest in the field has come along with the birth of social networking sites like Facebook and Twitter. Social media generate a substantial amount of opinionated data, also referred to as **user-generated content** (Moens *et al.*, 2014), that offers valuable insights into the public opinion online.

The accessibility to an ever-growing stream of opinionated data represents one side of the coin, the introduction of what we could call "noise" the other. Given the speed at which social media data are produced, their informal nature and the restrictions with respect to message length, these data are characterised by a rather informal language containing misspellings, grammatical errors, emoticons, abbreviations, slang, etc. (Barbosa and Feng, 2010). Since many natural language processing (NLP) tools have been developed for and trained on standard language, a severe drop in performance is observed when applying these tools to user-generated content (Eisenstein, 2013). Moreover, due to an inconsistent writing style, user-generated content is prone to increase feature sparseness, which complicates the classification task.

In this paper, we present a sentiment analysis pipeline for social media data exploiting a wide variety of information sources. We proceed to **joint optimisation** of the classifier by simultaneously performing feature selection and hyperparameter configuration. In a next step, we investigate to what extent the performance of our optimal sentiment classifier can be further enhanced by applying complex statistical machine translation (SMT)-based **normalisation** to the data prior to training. Although data noisiness has been widely acknowledged as one of the main challenges in the field (Liu, 2015), this is, to our knowledge, the first study to provide a qualitative and comparative analysis of a complex normalisation system and to experimentally test its benefits for an optimised sentiment classifier.

The remainder of this paper is structured as follows. Section 2 presents an overview of sentiment analysis research and text normalisation, while section 3 zooms in on the normalisation system used for this research. Section 4 describes our sentiment analysis architecture and details the experimental setup, after which the results are thoroughly discussed in section 5. Finally, section 6 highlights the conclusions and presents some directions for future research.

2. Related research

Traditional communication tools have increasingly given way to social media, which have become a rich source of opinionated data. To be able to analyse such an amount of data, automatic text processing techniques like sentiment analysis have become highly relevant.

2.1. *Sentiment analysis*

Natural language processing has many applications related to text analysis and initial text classification studies focused on extracting factual information from documents, which resulted in systems for automatic information retrieval (Salton, 1989), text summarisation (Nenkova, 2005), document classification (Finn *et al.*, 2002), topic modelling (Deerwester *et al.*, 1990) and so on. More recently, the research focus of the NLP community is increasingly geared towards the extraction of subjective information from text, thus introducing the field of *sentiment analysis* or the automatic classification of a document as positive, negative or neutral (i.e. when a positive and negative sentiment is expressed, or neither of them).

In its early days, sentiment analysis was mostly applied to newswire documents and movie reviews, seminal work on which has been done by Wiebe *et al.* (2004) and Pang and Lee (2008). The interest in the domain has thrived with the expansion of the Internet and the birth of social media, providing a large amount of opinionated data that is increasingly searched for by researchers, companies, politicians, and trend-watchers. As a result, sentiment analysis has become a major research area in NLP, which is reflected in survey articles by Liu (2015) and Mohammad (2016).

Two dominant approaches to sentiment analysis exist: (i) a **machine learning-based** approach and (ii) a **lexicon-based** approach. Machine-learning approaches are either supervised or unsupervised. Supervised methods include classifiers that are trained on labelled or annotated documents, whereas unsupervised methods infer the semantic orientation of a document without labelled data (e.g. using clustering or seed words). Semi-supervised approaches are a combination of both, for instance when sentiment lexicons are used, but no manually labelled data. Lexicon-based approaches make use of dictionaries of sentiment words to calculate a global sentiment score for a document. Both lexicon-based and machine-learning approaches present difficulties: while the former suffer from the varying and changing nature of language, machine-learning approaches generally require a large corpus of labelled documents to train a model. The two methods have been investigated extensively over the past years and this has resulted in a fair amount of benchmark data in the field. In machine learning, researchers have investigated and compared the performance of different classification and regression algorithms (Davidov *et al.*, 2010; Agarwal *et al.*, 2011), and more recently successful sentiment analysis approaches involve deep learning using neural networks (Severyn and Moschitti, 2015).

Sentiment analysis has also increasingly attracted researchers' interest in the framework of specialised shared tasks like SemEval, a series of evaluations of computational systems for tasks related to semantic text analysis. Datasets for training and testing are provided by the task organisers so that the participants' systems can be compared against each other. SemEval-2013 to -2017 workshops included a Twitter sentiment analysis task, which has resulted in state-of-the-art sentiment classifiers and a number of new resources, such as the NRC Hashtag Sentiment Lexicon (Mohammad *et al.*, 2013). Over the years, the shared task has attracted approximately 200 research teams, submitting together more than 300 sentiment analysis systems. Popular classification methods are Support Vector Machines (SVMs), Naïve Bayes, Conditional Random Fields (CRFs), and deep learning architectures. Often exploited features include bags-of-words (i.e. word or character n-grams), syntactic features based on part-of-speech (PoS) information or dependency relations, semantic features capturing modality and negation, sentiment lexicon features and user-based features. Twitter-specific features are often included to represent creative writing (e.g. punctuation, capitalisation, character flooding, emoticons and at-replies) (Nakov *et al.*, 2016). In short, recent research has successfully explored sentiment classification on Twitter, presenting classifiers with performance scores of up to F_1= 0.69 (Rosenthal *et al.*, 2017). Nevertheless, it has also unveiled a number of bottlenecks, one of them being the use of highly informal language or **data noisiness**.

2.2. *Handling "data noisiness"*

One of the larger and less-researched issues NLP is currently facing is *data noise* (Liu, 2015), which refers to non-standardised language variation. In fact, an inherent characteristic of social media data is its tendency to deviate from the linguistic norm, as it often contains misspellings, creative punctuation and inconsistent capitalisation, or has a poor grammatical structure. This hinders automatic text processing with traditional NLP tools that are trained on standard text and show a significant drop in performance when applied to social media data (Eisenstein, 2013). Among others, Ritter *et al.* (2011) demonstrated that the performance of existing tools for PoS tagging and chunking decreases when applied to tweets, and similar findings were reported by Liu *et al.* (2011) for named entity recognition.

As described by Han *et al.* (2013), one way to minimise the performance drop of such tools is to adapt or **retrain** them using social media data. Recent studies have introduced Twitter-specific tools for, among other tasks, PoS tagging (Gimpel *et al.*, 2011), and named entity recognition (Ritter *et al.*, 2011). A drawback of this strategy is that it implies a process of continuous retraining of each individual tool to make it robust to changes in language use. According to the researchers, another strategy to handle non-standard language is **text normalisation**, which transforms noisy or non-standard text into its standard form. Three dominant approaches to text normalisation exist, referred to as the spell-checking, the machine translation and the speech recognition metaphors (Kobus *et al.*, 2008). The former metaphor refers to normal-

isation as a word-per-word correction primarily targeting out-of-vocabulary (OOV) words. The translation metaphor refers to normalisation as a machine translation task where the noisy text has to be "translated" into its standard form. Finally, the speech recognition metaphor alludes to the analogy between noisy and spoken language, assuming that noisy content like SMS messages tends to be closer to oral expressions than to written text. For a comprehensive overview of the different normalisation approaches, we refer to Kobus *et al.* (2008).

Although various normalisation systems have been developed and have proven successful (De Clercq *et al.*, 2013; Pettersson *et al.*, 2013; Schneider *et al.*, 2017), most approaches to sentiment analysis only incorporate shallow normalisation such as spell-checking mechanisms or rule-based normalisation without evaluating the impact of normalisation on the classification performance (Roy *et al.*, 2011; Haddi *et al.*, 2013; Sharma *et al.*, 2015). In the present study, we therefore investigate the impact of an SMT-based normalisation system as preprocessing for our optimised sentiment classifier.

3. Normalisation of user-generated content

As discussed in the previous sections, data noisiness may severely undermine the accuracy of NLP tasks including tokenisation, lemmatisation and PoS tagging. Moreover, it may hinder the effectivity of bag-of-words or lexicon-based features due to increased sparseness. The negative effects of data noise are well-known, but only few studies have applied normalisation as a preprocessing step for text mining tasks. To our knowledge, the actual impact of complex normalisation on sentiment classification performance has not been investigated sufficiently. Mosquera and Moreda (2013) did tackle the problem and showed a 6% polarity classification improvement after applying normalisation to tweets, but, in contrast to the current approach, they implemented a normalisation system based on dictionaries, rather than a statistical normaliser trained on user-generated data.

We hypothesise that normalisation helps to reduce feature sparseness, as was shown by Desmet (2014), and consequently improves the coverage of sentiment lexicons that are often used to construct features for this task. To this purpose, we applied SMT-based normalisation (De Clercq *et al.*, 2013), the underlying idea of which is to perceive normalisation as a translation process from noisy text into normalised standard text in the same language. An SMT system basically consists of two subsequent models: a **translation model** that is trained to find the right target words given the source words and a **language model** that ensures the translated words come in the right order. Applying SMT to text normalisation can be done at different levels of granularity (i.e. at the token or character level). The advantage of working at the token level is that the high-frequency words and abbreviations can be translated in context, which outperforms a dictionary look-up (Raghunathan and Krawczyk, 2009). A character-based translation module can translate non-standard words that were not

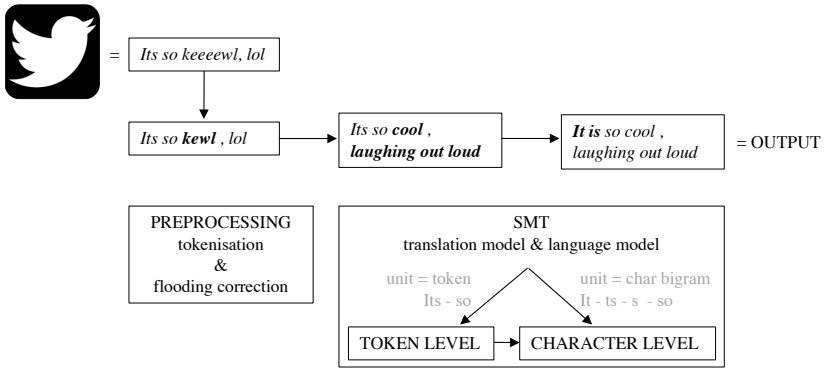

Figure 1. *Illustration of the normalisation process. After preprocessing the tweet, it is normalised at the token level, followed by a normalisation at the character level.*

seen in the training set as it learns patterns of character sequences rather than entire words, which makes the system more robust (Li and Liu, 2012).

For the current research, we propose an SMT-based approach and make use of a system developed by De Clercq *et al.* (2013) for text normalisation of (initially Dutch) user-generated content. The system is based on statistical machine translation and makes use of the SMT system Moses (Koehn *et al.*, 2007). As a target corpus for the language model, which was built with KenLM (Heafield, 2011), parts of the English Open Subtitles Corpus [1] were used, since the data in this corpus are user-generated. The training data for the normalisation system comprises texts from the social networks site ASKfm [2], YouTube comments (Dadvar *et al.*, 2014) and a subset of the Twitter corpus (TWE) compiled by Xue *et al.* (2011). The final corpus comprises about 60,000 tokens and has been manually normalised using annotation guidelines (De Clercq *et al.*, 2014). The system performance was evaluated on three different types of user-generated content, being message board posts (SNS), text messages (SMS), and tweets (TWE). With F_1 scores of up to 0.65 for all three genres, the system has demonstrated its robustness across different genres of user-generated content.

The system, the process of which is depicted in Figure 1, works as follows: prior to performing SMT, the noisy text is tokenised using the Twitter tokeniser by Gimpel *et al.* (2011). Subsequently, a cascaded approach is taken: the system first processes a noisy tweet at the token level to find frequent abbreviations (such as "lol" for "laughing out loud"), after which the output is split into character sequences for normalisation at the character level. The latter step should allow to better resolve character transpo-

1. http://www.opensubtitles.org.
2. http://ask.fm.

sitions, but also problems across the token level, like fusions such as “its” for “it is”. As mentioned earlier, character-based translation models also better generalise since they can learn productive alternations and correct them in words that do not occur in the training data. An important decision to make when applying such a cascaded approach is the level of granularity at the character level, for which we tested unigrams and bigrams.

	original	**normalised**
SMS	Ok then later i **msg u** where i am.	Ok then later i **message you** where i am.
	yah, **wed** i think. But i **noe she not** free on **wed**.	**yah**, **wed** i think. But i noe not free on **wed**.
Twitter	@Manlyboyze I would put Gillard **2nd** best behind Hawke, Rudd **3rd**, Keating 4th, then all the rest, then Howard last #Showdown	@Manlyboyze I would put Gillard **second** best behind Hawke, Rudd **third**, Keating 4th, then all the rest, then Howard last #Showdown

Table 1. *Corpus examples before and after normalisation.*

We empirically verified which cascaded approach (i.e. token+character unigrams or token+character bigrams) works best for the research presented here by selecting 100 random instances from the English SemEval Twitter training corpus (Rosenthal *et al.*, 2014). These 100 tweets (comprising 2,185 tokens all together) were manually normalised by two trained linguists who performed the task independently. In a next phase, a few conflicting normalisations were discussed and resolved, which resulted in a total of 206 tokens that were normalised. Next, the dataset was automatically normalised using two different flavours of the normalisation system and we thus compared the output of the cascaded token+character unigram approach (system A) with that of the cascaded token+character bigram approach (system B). We found that the systems perform comparably in terms of recall, given that system A resolved 41% of the normalisations, whereas system B resolved 40%. When looking at the precision of both systems, however, we found that system A hypernormalised more tokens, which often led to strange additions. As such, precision of the system was 61%, as opposed to 86% of system B. We therefore decided to use system B or the cascaded token+character bigram module for the normalisations performed in the framework of this paper. This finding is in line with previous experiments on Dutch data which revealed that the best performance is indeed achieved with a cascaded bigram approach (De Clercq *et al.*, 2013).

4. Sentiment analysis architecture

This section presents the experimental corpus and feature engineering process, after which the feature filtering and normalisation experiments are described.

4.1. *Data preprocessing and feature engineering*

For the experiments we made use of a training and a held-out test corpus, both containing English social media data. The corpora were distributed in the framework of the SemEval-2014 shared task on sentiment analysis (Rosenthal *et al.*, 2014) and consist of 11,338 and 8,987 instances, respectively (Table 2).

train corpus	**test corpus**		
Twitter	Twitter	SMS	blog
11,338	5,752	2,093	1,142

Table 2. *Number of instances in the experimental corpus.*

The training corpus contains merely tweets, whereas the test corpus comprises a variety of user-generated content, including tweets (regular and sarcastic), text messages (SMS), and LiveJournal [3] blog posts. All instances in the corpora are assigned one out of three class labels, being *positive*, *negative*, and *neutral*, which represent 37%, 16% and 47% of the training data. The class distributions in the test set are comparable.

Prior to extracting features, all tweets in the *norm* and *not norm* corpus were tokenised and PoS tagged using the Carnegie Mellon University Twitter NLP Tool (Gimpel *et al.*, 2011). The tag list comprises regular PoS tags (e.g. "N" for noun, "V" for verb), as well as Twitter-specific ones (e.g. "#" for hashtags, "E" for emoticons). The following paragraphs describe the different features that were extracted to provide the model with relevant information for the task.

Bag-of-words features (BoW): features that represent each message as a "bag" of its words or characters. Unigrams, bigrams and trigrams were extracted at the token level and trigrams and fourgrams at the character level (without crossing token boundaries). N-grams that occurred only once in the training corpus were discarded, which decreased the total number of n-gram features by 80% and the entire feature space by about 50%.

Post length: numeric feature indicating the tweet length in tokens.

Word-shape features: numeric and binary features including (i) the number of tokens with character (e.g. *yaaaay*) and (ii) punctuation flooding (e.g. *?*!!*), (iii) whether the last token of the tweet contains punctuation, (iv) the number of tokens in uppercase and (v) the number of hashtags in the tweet. All the numeric features were normalised by dividing them by the tweet length.

3. http://www.livejournal.com.

Syntactic features:

– **Part-of-Speech (PoS) features:** four features for each one of the 25 tags in the PoS tagset, indicating (i) whether the tag occurs in the tweet, (ii) whether the tag occurs zero, one, or two or more times, (iii) the absolute and (iv) relative frequency of the tag. It is important to note that the Twitter-specific PoS tagger distinguishes hashtags that are part of the tweet content from those with a "hashtag function". For instance, in "Foreign commentator award at the #commentawards goes to Marie Colvin", *#commentawards* is tagged as a noun, while in "I hope I get to meet Carmelo Anthony at the Knicks Rally #Hopeful", *#Hopeful* is tagged as a hashtag.

– **Dependency relation features:** four binary features for every dependency relation found in the training data (example 1). The first feature indicates the presence of the lexicalised dependency relations in the test data (hm-lex). For the remaining features, the dependency relation features are generalised in three ways, as proposed by Joshi and Penstein-Rosé (2009): by backing off the head word to its PoS tag (h-bo), the modifier to its PoS tag (m-bo), and both the head and modifier (hm-bo). The dependency parser we made use of is not adapted to Twitter data, hence hashtags are treated like other words.

(1) I had such a **great time** tonight that I've decided to keep celebrating! → hm-lex: (*time*, *great*), h-bo: (N, *great*), m-bo: (*time*, A), hm-bo: (N, A)

Named entity features: four features indicating the presence of named entities in a tweet: one binary feature and three numeric features, indicating (i) the number of NEs in the tweet and (ii) the number and (iii) frequency of tokens that are part of a NE.

PMI features: two numeric features based on PMI (*pointwise mutual information*) obtained from (i) word-sentiment associations in the training data, and (ii) an existing PMI lexicon (Mohammad *et al.*, 2013). A positive PMI value indicates positive sentiment, a negative score indicates negative sentiment. The higher the absolute value, the stronger the word's association with the sentiment. PMI values were calculated by subtracting a word's association score with a negative sentiment from the word's association score with a positive sentiment, as shown by the following equation: $PMI(w) = PMI(w, positive) - PMI(w, negative)$.

Sentiment lexicon features: four sentiment lexicon features were implemented to capture the semantic orientation of a tweet. We consulted existing lexicons including AFINN (Nielsen, 2011), General Inquirer (GI) (Stone *et al.*, 1966), MPQA (Wilson *et al.*, 2005), the NRC Emotion Lexicon (Mohammad and Turney, 2010), Liu's opinion lexicon (Hu and Liu, 2004), Bounce (Kökciyan *et al.*, 2013) and a manually created emoticon list based on the training data. Four features were extracted per lexicon: the number of positive, negative and neutral lexicon words averaged over text length, and the overall polarity (i.e. the sum of the values of the identified sentiment words). These features were extracted by looking at (i) all the tokens in the instance and (ii) hashtag tokens only (e.g. *win* from *#win*). Negation cues were considered by

flipping the polarity of a sentiment word if it occurred within a window of three words left or right to a negation word from a manually composed list (e.g. “neither”, “without”, “cannot”). To minimise ambiguity, a word was only attributed a sentiment value if its PoS tag matched that of the dictionary entry (when available).

4.2. *Experimental setup*

Our two main research objectives are (i) to build a sentiment classifier that makes optimal use of a varied feature set (cf. section 4.1) and (ii) to investigate the impact of lexical normalisation when included as a preprocessing step (cf. section 4.2.3). For the experiments we made use of a support vector machine as implemented in the LIBSVM library (Chang and Lin, 2011), since the algorithm has been successfully implemented with large feature sets and its performance for similar tasks has been recognised (Zhu *et al.*, 2014). Prior to constructing models, all feature vectors were scaled so that each variable fits in the range [0, 1]. As the evaluation metrics, we report accuracy, and macro-averaged precision, recall and F_1 score. Macro-averaging was preferred over micro-averaging as it attributes equal weight to the different classes in the evaluation (Sokolova and Lapalme, 2009). We recall that these classes are positive, negative and neutral (cf. section 4.1).

4.2.1. *Baselines*

Three standard baseline classifiers were implemented against which to compare the models’ performance, being the majority and random class baseline and a unigram-based model (w1gr) relying solely on word unigram features. For this model, the LIBSVM classifier was applied in its default parameter settings. Evaluation was done using ten-fold cross validation.

baseline	**accuracy**	**precision**	**recall**	**F_1**
majority	46.97%	15.66%	33.33%	21.30%
random	38.91%	33.58%	33.58%	33.58%
w1gr	46.97%	15.66%	33.33%	21.30%

Table 3. *Cross-validated results for the majority, random and unigram baseline.*

As can be deduced from the table, scores for the baselines are low. The (unoptimised) unigram baseline scores equally to the majority baseline as it consistently predicts the neutral class, which demonstrates the importance of hyperparameter optimisation for our LIBSVM classifier.

4.2.2. *Building an optimal sentiment classifier exploiting a rich feature set*

It has been shown that sentiment classification benefits from a variety of information sources including bags of words, syntactic features, sentiment lexicon features, negation and modality clues, and so on (Zhu *et al.*, 2014). Implementing such a rich

feature set seems promising, since many individual features may have a substantial predictive power when they are combined. However, it is probable that not all individual features are equally informative for the decision-making algorithm. For instance, bag-of-word features have shown to perform well for sentiment analysis (Rosenthal *et al.*, 2015), but easily suffer from sparseness as the corpus size increases (Saif *et al.*, 2012). High-dimensional data with a large number of features may include irrelevant and redundant information, which could degrade the performance of learning algorithms (Yu and Liu, 2003). hat is why in our first round of experiments, we disentangle which types of information sources described in section 4.1 lead to optimal classification performance by means of feature filtering and hyperparameter optimisation using a genetic algorithm (see further). It is important to note that, in this experimental round, normalisation is not included as a preprocessing step.

Dimensionality reduction of the feature space is done in two phases, including (i) feature filtering based on information gain (Daelemans *et al.*, 2009) and (ii) wrapped feature selection.

(i) **Feature filtering** is done based on information gain (IG). IG is the difference in entropy, i.e. the uncertainty about a class label given a set of features when the feature is present or absent in a feature vector representation. Given the considerable number of individual features, we decided to apply feature filtering prior to joint optimisation using a wrapper method. We experimentally determined a threshold of 0.001, meaning that features with an IG value below 0.001 were discarded. This reduced our initial feature space by approximately 99.5% (i.e. from 430,980 to 1,852 features).

(ii) **Wrapper methods**, as opposed to statistical feature filtering (see (i)), conduct a search for informative features using the induction algorithm itself as part of the evaluation function (Kohavi and John, 1997). The selected features are the ones that have the most predictive power given the particular task and classification algorithm. The advantage of wrapped feature selection is that it tests features in combination to detect the possible interactions between them. An important disadvantage of wrapper methods is that they are computationally expensive for data with numerous features (Chandrashekar and Sahin, 2014). Two measures were taken to prevent the search space from becoming too large. First, the feature space was filtered using information gain, as explained in the previous paragraph. Second, the remaining individual features were combined into feature groups for feature selection, since 1,852 individual features would increase the search space considerably due to combinatorial explosion.

Another challenging but necessary (see the results of the unoptimised unigram model in Table 3) task when applying machine learning algorithms is defining good hyperparameter settings as it allows a classifier to better fit the training data. Out of the variety of hyperparameters that can be set for a LIBSVM classifier, we chose to optimise kernel-specific settings including t (the kernel type), d (the kernel function degree), g (the kernel function gamma), and the classification cost value, C. Hoste (2005), among other researchers, demonstrated the importance of *joint optimisation*, meaning that feature- and hyperparameter selection are performed at the same time

so that their mutual influence can be evaluated. To this purpose, we made use of the Gallop (Genetic Algorithms for Linguistic Learner Optimisation) toolbox (Desmet *et al.*, 2013). Optimisation was done using k-fold cross validation on the training set and the number of individuals to be tested per generation was set to 100. Given the size of the training corpus, k was set to 3. Both the feature groups and hyperparameters were defined so as to maximise macro-averaged F_1 to assign equal weight to each class in the evaluation.

4.2.3. *Normalisation experiments*

We hypothesised earlier that data noise could (i) increase sparseness in the feature space and (ii) lower the coverage of sentiment lexicons. On the positive side, however, character flooding or extensive capitalisation may provide hints for automatic sentiment analysis. To assess the actual impact of normalisation on this task, we compared the performance of our most optimal sentiment classifier when trained on not-normalised versus normalised input. Before presenting the experimental results, we already provide some insights into the qualitative analysis of the input data before and after normalisation.

Related to the first hypothesis, we observed that token n-grams in the normalised corpus were more sparse: 54,326 token n-grams versus 53,458 in the not-normalised corpus. A qualitative analysis revealed that, while the normaliser decreased sparseness by normalising spelling variations like "yesssss" and "yesss" to "yes", it increased sparsity through (i) correct normalisations that resulted in more tokens than the original word (e.g. "btw" $\rightarrow$ "by the way", "gonna" $\rightarrow$ "going to"), but also through strange operations and hypercorrections (e.g. "#Illini"$\rightarrow$ "#I willini"). Also, given that n-grams with a frequency less than two in the training corpus were discarded during preprocessing, it is likely that normalisation increased the number of tokens added to the bag-of-words. For instance, "Jerry's" and "Jerrys" both occur once in the corpus and were discarded from the bag-of-words in the not-normalised corpus, but during normalisation, "Jerrys" was corrected to "Jerry's", and occurring more than once now, the word was added to the bag-of-words. Normalisation did, however, cause a slight decrease in sparsity in the character-level n-grams (–1%) and the dependency features (–0.5%).

Regarding our second hypothesis, we observed that after normalising the input text, the coverage of the sentiment lexicons increased by 0.22%. This would indicate that normalisation has a rather limited effect on sentiment-bearing words, which could be because (i) their noise cannot be resolved by the normaliser or (ii) they barely need correction. The results of a qualitative analysis of the corpus are in favour of the latter assumption, revealing that syntactic words (e.g. "wiv u" $\rightarrow$ "with you", "ima" $\rightarrow$ "I am going to") appeared much more prone to deviant spelling in our corpus than sentiment-bearing words.

5. Results

In section 4.2.2 we described the steps towards our most optimal classifier. Now we discuss its results and take a closer look at the selected features and hyperparameter settings (sections 5.1 and 5.2.1). Furthermore, we investigate the impact of normalisation as preprocessing on the classification performance. To this end, a second optimised classifier was built after the SMT-based normalisation system was applied to the dataset, leading to a more standard text input corpus. We then discuss the results of our optimised classifier before and after normalisation (section 5.2). Finally, this section zooms in on the performance of our optimised classifiers on a held-out test set containing different genres of user-generated content (section 5.2.2).

5.1. *The effect of classification optimisation*

Observing poor results by the unoptimised LIBSVM classifier, and hypothesising that not all individual features contribute equally to the classification performance (cf. section 4.2.2), we investigate the effect of **joint optimisation**. Table 4 compares the results of two classifiers before and after joint optimisation. A unigram-based model is compared to that of a rich-feature-based model to investigate the benefits of a rich feature set for this task.

model features	optimised?	accuracy	F_1	precision	recall
1) unigram BoW	-	46.97%	21.30%	15.66%	33.33%
2) rich feature set	-	46.97%	21.30%	15.66%	33.33%
3) unigram BoW	✓	67.87%	63.00%	66.98%	61.58%
4) rich feature set	✓	**79.20%**	**75.29%**	**77.55%**	**73.94%**

Table 4. *Cross-validated results for the unigram and rich feature-based models, before and after joint optimisation.*

As can be deduced from the table, when applied in its default parameter settings, the classifier consistently predicts the majority class, independently of the features that are used to construct the model (cf. systems 1 and 3). This demonstrates once more (cf. Table 3) that LIBSVM's default kernel type (RBF) is sensitive to a high-dimensional feature space if no suitable hyperparameter settings are defined (Hsu *et al.*, 2003). Joint optimisation clearly pays off, as it increases the performance of the unoptimised classifiers by 42% and 54%. The results further demonstrate that, when optimised, the classifier benefits from a rich feature set, yielding an F_1 score of 75.29%, as opposed to F_1= 63% when only word unigram features are used.

5.2. *The effect of normalisation on sentiment classification*

Having in place an optimised sentiment classifier, we now investigate the impact of integrating SMT-based normalisation as preprocessing on the classification performance. Table 5 presents the three-fold cross-validation results obtained for both optimised models, trained on normalised and not-normalised tweets (hereafter referred to as the "norm" and "not norm" setup). Similarly to the setup without normalisation as preprocessing, optimisation of the classifier involves IG-based feature filtering as well as joint feature selection plus hyperparameter optimisation using Gallop. Feature filtering decreased the feature space by approximately 99.5% (from 400,872 to 2,002 features), which is similar to the feature space reduction we observed for the corpus without normalisation (cf. section 4.2.2).

	overall score				*F_1 score per class*		
	accuracy	**F_1**	**precision**	**recall**	**F_1(pos.)**	**F_1(neg.)**	**F_1(neu.)**
TWE not norm	**79.20%**	**75.29%**	**77.55%**	**73.94%**	**82.78%**	61.22%	**81.87%**
TWE norm	77.77%	74.17%	76.00%	73.02%	80.39%	**61.39%**	80.73%

Table 5. *Cross-validated results of the optimised classifier with and without normalisation as preprocessing.*

As shown in Table 5, with F_1 scores of 75.29% and 74.17%, the optimised *not norm* and *norm* models outperform the baselines and unoptimised models with more than 50%. Surprisingly, the model constructed from the raw (i.e. not normalised) corpus slightly outperforms the *norm* model. As described in section 3, the precision of our normaliser is high (i.e. 86%), hence it is unlikely that its error rate exceeds the amount of noise in the corpus. Results on the held-out test set (see further) could, however, provide more insights into the performance of both models. The scores per sentiment class indicate that both classifiers perform better on the positive and neutral classes compared to the negative class. This could be explained by the lower number of negative class instances in the training data (cf. section 4.1).

In what follows, we evaluate the performance of the *norm* and *not norm* models on the held-out test set comprising a variety of user-generated content genres. Assuming that they each contain a different level of noise, we take a closer look at the classifier performance for each genre. Before discussing the results, we zoom in on the feature groups that were selected during joint optimisation.

5.2.1. *Selected features in the normalised and not-normalised setups*

In this section, we scrutinise which feature groups and hyperparameter settings were chosen after optimisation of the *norm* and *not norm* models by means of the genetic algorithm Gallop (Desmet *et al.*, 2013). We recall that the feature groups were created from the remaining features after filtering the feature space using information gain (cf. section 4.2.2). For the *not norm* model, all features were grouped into 36 different feature groups depending on their nature (e.g. token unigrams, token bigrams,

binary NE features, NRC sentiment lexicon features, and so on). Since all hm-bo dependency features (cf. section 4.1) were discarded after feature filtering using Information Gain [4], only 35 feature groups were retained for the *norm* setup. Hence, respectively 36 and 35 feature groups were considered for wrapped feature selection in the *not norm* and *norm* setup. During the optimisation, different combinations of hyperparmeter settings and feature group activations were evaluated by measuring the classifier performance through a three-fold cross validation experiment. At the end of an optimisation run, the highest fitness score indicated the classifier performance given a particular combination of hyperparameter settings and feature group activations. For both our models, this score is shown in Table 5.

As this score may be shared by other individuals with different feature group activations or hyperparameters (for a detailed explanation, see Desmet (2014)), it is advisable to not only discuss the individual with the highest fitness score, but also a number of individuals that obtained a comparable fitness score. For the analysis of the selected features and hyperparameter settings, we therefore discuss the k-nearest fit solution set as proposed by Desmet (2014), i.e. by rounding the scores to three decimals and selecting the top three fitness scores out of these.

Based on the top fitness scores, we carefully analysed the hyperparameter settings and selected the feature groups that were activated in the seven and six best individuals of the TWE *norm* and *not norm* models, respectively. In section 4.2 we defined which LIBSVM parameters are relevant to this task. The output of the optimisation process revealed that the linear kernel (t=0) was always selected as the most suited kernel type for the *norm* model, whereas the sigmoid kernel (t=3) was selected for the *not norm* model. In effect, when looking at the results of the optimised models, we see a substantial improvement over the results obtained with the classifier in its default kernel configuration (Table 5), demonstrating that defining a suited kernel is instrumental to LIBSVM's good performance with large feature spaces. For the normalised model, the optimal cost value C of the k-nearest individuals varied between 0.25 and 1, where it varied between 256 and 1,024 for *not norm*. The d parameter [5] is not relevant here given that a linear and sigmoid kernel were defined. The γ parameter [6] is not relevant when using a linear kernel, but it was set to an optimal value of 0.0039 for the *not norm* model.

Figure 2 visualises the activation of the different feature groups in the three-nearest fit individuals for both models. Feature groups that are consistently retained in the fittest individuals can be considered the most important given the nature of the classification task. In both figures the right column contains information about the selection status of the feature group, represented as a percentage and with a colour range. The

4. In the same way, for both *norm* and *not norm*, sentiment features based on hashtag tokens were discarded for the Bounce and SemEval (se) lexicons after IG-filtering. This makes intuitive sense as both are emoticon lexicons and do not consider regular words.

5. d is the degree of freedom of the polynomial kernel.

6. Intuitively, the gamma parameter defines how far the influence of a single training example on the model reaches.

Feature group activation NOT NORM	
postLength	100%
countFloodedTokens	0%
countCapitalizedTokens	66.67%
countFloodedPunctuationTokens	100%
countHashtags	100%
punctuationLastToken	66.67%
token1gramFeatures	100%
token2gramFeatures	100%
token3gramFeatures	50%
character3gramFeatures	0%
character4gramFeatures	100%
dependency_hm-lex	0%
dependency_h-bo	0%
dependency_m-bo	0%
PoSBinaryFeatures	100%
PoSTernaryFeatures	83.33%
PoSAbsoluteFeatures	100%
PoSFrequencyFeatures	100%
NEBinary	66.67%
NEAbsolute	66.67%
NETokenAbsolute	66.67%
NETokenFrequency	66.67%
nrc	33.33%
gi	33.33%
liu	33.33%
mpqa	100%
afinn	100%
bounce	66.67%
se	33.33%
nrc-hashtokens	100%
gi-hashtokens	66.67%
liu-hashtokens	33.33%
mpqa-hashtokens	100%
afinn-hashtokens	83.33%
nrc-PMIScore	66.67%
se-PMIScore	100%

Feature group activation NORM	
postLength	71.43%
countFloodedTokens	71.43%
countCapitalizedTokens	0%
countFloodedPunctuationTokens	57.14%
countHashtags	57.14%
punctuationLastToken	71.43%
token1gramFeatures	71.43%
token2gramFeatures	100%
token3gramFeatures	28.57%
character3gramFeatures	28.57%
character4gramFeatures	100%
dependency_hm-lex	0%
dependency_h-bo	0%
dependency_m-bo	28.57%
dependency_hm-bo	0%
PoSBinaryFeatures	28.57%
PoSTernaryFeatures	85.71%
PoSAbsoluteFeatures	100%
PoSFrequencyFeatures	100%
NEBinary	100%
NEAbsolute	0%
NETokenAbsolute	71.43%
NETokenFrequency	0%
nrc	100%
gi	71.43%
liu	0%
mpqa	100%
afinn	100%
bounce	0%
se	0%
nrc-hashtokens	85.71%
gi-hashtokens	85.71%
liu-hashtokens	85.71%
mpqa-hashtokens	0%
afinn-hashtokens	85.71%
nrc-PMIScore	85.71%
se-PMIScore	100%

Figure 2. *Gallop feature group activation of the models with (*norm*) and without (*not norm*) normalisation as preprocessing.*

higher the percentage of individuals where the feature group is selected, the darker the tone. The figures are split in such a way that feature sets of the same nature or related feature sets are grouped together. The upper part comprises the word-shape features, the post length, and hashtag count. The second part shows the bag-of-word features at the token and character level. The third part contains the dependency relation features,

the PoS tag features and the named entity features. After that, the sentiment lexicon features are presented per lexicon (NRC, General Inquirer (gi), SemEval (se), etc.), for regular and hashtag tokens. At the bottom of the table are the PMI-based features. Overall, we notice that more features groups are retained in the *not norm* setup (31 out of 36) than in the *norm* setup (27 out of 37). If we zoom in on the differences between *norm* and *not norm*, we observe that **social media-specific features**, i.e. features that capture creative writing like punctuation flooding and capitalisation, are more often activated in the *not norm* than in the *norm* model. Interestingly, this is not the case for *countFloodedTokens*, a feature that captures the number of tokens with character flooding, which is removed during normalisation. A qualitative analysis revealed, however, that not all types of flooding were removed during normalisation. As such, important features were derived from words like "yummmm" and "xxxxx" where the flooded characters are at the end of the word instead of in the middle. These words were, as opposed to "yaaay" and "looooooll", not corrected by the normaliser. When looking at the sentiment features, we see that information provided by the MPQA and AFINN lexicons results in good features for both models, since they show a 100% activation across the best individuals. The AFINN lexicon also shows a high activation when applied to hashtag tokens in both models. This is in line with the work by Özdemir and Bergler (2015), revealing that AFINN outperforms other popular lexicons for sentiment analysis tasks.

Overall, the top three most important feature groups (i.e. with the highest overall activation) for the *not norm* model include PMI-based features (i.e. nrc-PMIScore and se-PMIScore), sentiment lexicon features (i.e. nrc-hashtokens to afinn-hashtokens) and word-based features (i.e. postLength to punctuationLastToken). For the *norm* model, the first and second most important feature groups (i.e. PMI and sentiment lexicon features) are the same, but the third most important group are n-gram features (i.e. token1gramFeatures to character4gramFeatures). The analysis further reveals that features based on dependency relations in the training data are the least informative, given that only one out of four dependency features (i.e. dependency m-bo) is activated in *norm*, but no dependency features were activated in the fittest *not norm* models.

5.2.2. *Results on the held-out test set*

In this section, we report the results of our optimised sentiment classifiers (i.e. trained on our corpus with and without normalisation as preprocessing) on a varied held-out test set. The test set has not been part of the corpus during training and optimisation and therefore allows to draw conclusions about the robustness of our system. It was distributed for the SemEval Task 9 (Rosenthal *et al.*, 2014) and consists of 8,987 messages from a variety of genres including Twitter (64%), SMS (23%), and blogs (13%). Table 6 displays the scores obtained by the two models. As evaluation metrics, we report accuracy and (macro-averaged) precision, recall and F_1 score.

We can observe that overall, the best scores are obtained with the *norm* model, reaching an F_1 score of 68.90% on the full test set. This is opposite to our findings in the cross-validation experiments (cf. Table 5) and indicates that the *norm* model

	overall score				*F_1 score per class*		
	accuracy	**F_1**	**precision**	**recall**	**F_1(pos.)**	**F_1(neg.)**	**F_1(neu.)**
TWE not norm	70.26%	68.23%	**73.02%**	67.22%	66.91%	**63.40%**	74.38%
TWE norm	**70.94%**	**68.90%**	72.55%	**67.80%**	**69.22%**	62.95%	**74.54%**

Table 6. *Results for each optimised model on the held-out test set, for all classes and for each class label separately.*

is more robust, allowing to perform better on unseen data, even when it contains different genres of UGC. We observe that both systems perform well on all three classes, although they show an important difference in distribution in the training data (cf. section 4.1). The negative class, for instance, only represents 16% of the training corpus. We can conclude from this that our optimisation strategy to take macro-averaged F_1 score as fitness criterion during optimisation pays off. Judging from the raw performance results, sentiment classification of user-generated content benefits from lexical normalisation as a preprocessing step. In a next step, we investigated whether the results of our normalised system are **significantly** better than those of the not-normalised system. To this end, we applied a t-test (α= 0.05) after bootstrap resampling (Noreen, 1989). More specifically, we drew samples (n= 5,000) with replacement from the output of each system (i.e. the classified instances) and of equal size of the test set (n= 8,987). For each resample, macro-averaged F_1 score was calculated and we subsequently applied a paired samples t-test to compare the mean scores for both systems, which revealed a significant difference ($p< 0.001$) between the results of the *norm* and *not norm* models.

In the above paragraphs, we demonstrated the benefits of normalisation as preprocessing for our sentiment classifier tested on an unseen and varying corpus of user-generated content. In what follows, we take a closer look at both models' performance for each data genre in the test set, the results of which are presented in Table 7.

	SMS2013	**TWE2013**	**TWE2014**	**TWE2014 sarc**	**LJ2014**
	2,093 inst.	3,813 inst.	1,853 inst.	86 inst.	1,142 inst.
TWE not norm	**71.57%**	66.89%	65.73%	46.38%	68.63%
TWE norm	70.31%	**68.26%**	**67.37%**	**50.42%**	**69.67%**

Table 7. *Macro-averaged F_1 scores for each optimised model per genre in the held-out test set.*

Interestingly, the results indicate that both models, which are trained on Twitter data (TWE), perform best on the SMS and LiveJournal genres. Not surprisingly, Twitter sarcasm seems to be the most difficult genre, but it also benefits most from the model where normalisation is included as preprocessing. The figures indicate that the *norm* model performs best for all genres except SMS. In fact, a qualitative analysis revealed that the genre is much more noisy than Twitter (as can be observed in the examples in Table 1 and which was also shown by De Clercq *et al.* (2013)). Hence,

it is not surprising that the *not norm* model better fits the SMS genre, as compared to *norm*.

It is noteworthy, however, that the performance of the normaliser on the SMS set is comparable to its performance on Twitter. In fact, a manual normalisation of 50 instances from the SMS test set showed that recall of the system was 39% and precision 87%, which is comparable to its performance on tweets (cf. section 3). To better evaluate the benefits of normalisation for sentiment analysis on other genres than Twitter, further experiments are necessary where such other genres are included in the training corpus of our sentiment model. In sum, the results in the table indicate that the *norm* model outperforms *not norm* on unseen data, and it generalises well to other data genres (e.g. LiveJournal) or variations (e.g. sarcastic tweets), except SMS.

As the experiments were run using SemEval data, we also compared the performance of our sentiment classifier in the light of the SemEval shared tasks on sentiment analysis. We focus particularly on the SemEval-2014 T9 (Rosenthal *et al.*, 2014) run in which we participated and where participants were required to classify (social media) messages as positive, negative or objective/neutral. Table 8 presents the scores of TeamX (Miura *et al.*, 2014), the Task 9 winning team, and compare them to the results of our two models (i.e. *LT3 norm* and *LT3 not norm*). The TeamX system is based on a supervised text categorisation approach by means of logistic regression. The features that are exploited include bags-of-words, sentiment lexicon features, cluster features, and word sense features. As preprocessing steps the authors included rule-based normalisation, spelling correction, PoS tagging, word sense disambiguation, and negation detection.

It is important to note that Table 8 displays scores obtained by two different evaluations. Firstly, we report the scores obtained by TeamX and LT3 on the three sentiment classes, being positive, negative and neutral. Secondly, we present the results of the SemEval-2014 competition, which scored the systems solely by their performance on the positive and negative class (Rosenthal *et al.*, 2014).

	SMS2013 2,093 inst.	**TWE2013** 3,813 inst.	**TWE2014** 1,853 inst.	**TWE2014sarc** 86 inst.	**LJ2014** 1,142 inst.	**Full test** 8,987 inst.
TeamX	53.62%	**71.72%**	**69.34%**	**55.06%**	58.11%	65.40%
LT3 (not norm)	**71.57%**	66.89%	64.73%	46.38%	68.63%	68.23%
LT3 (norm)	70.31%	68.26%	67.37%	50.42%	**69.67%**	**68.90%**
Official results SemEval-2014 Task 9						
TeamX	57.36%	72.12%	70.96%	56.50%	69.44%	65.63%
LT3	64.78%	65.56%	65.47%	47.76%	68.56%	60.60%

Table 8. *Comparative results on sentiment classification: the SemEval 2014 winning team versus the current system (with and without normalisation as preprocessing) for the classification of pos/neg/neu sentiment and the official SemEval-2014 Task 9 scores for the classification of pos/neg sentiment.*

As shown in Table 8, with a top F_1 score of 68.90% obtained by our *norm* model, our system compares favourably with the top-performing TeamX system, which obtained an F_1 score of 65.40% on the complete test set. Looking at the different genres in the test set, we observe that TeamX obtained the best results on the Twitter genre, whereas our system performed particularly well on the SMS and LiveJournal (LJ) genre. This demonstrates our system's robustness to data genres other than Twitter, while TeamX' system was particularly tuned to Twitter data, making use of eleven different types of n-gram features (contiguous and not-contiguous, token-based and character-based, both with different levels of granularity) and word clusters generated from a 56M Twitter corpus [7].

Looking at the official SemEval-2014 Task 9 results, we observe that the TeamX system outperformed ours on the positive and negative sentiment class, except for SMS2013. Both systems (i.e. TeamX and LT3) were, however, trained on the three sentiment classes. A possible explanation is that the TeamX system integrated weighting of the predicted labels to handle with the class imbalance in the test set (Miura *et al.*, 2014). Overall, comparing these scores with the performance of our current system clearly demonstrates the beneficial effects of classifier optimisation and normalisation, which allowed us to outperform the Task 9 wining team with 3.5 points (i.e. 68.90% vs. 65.40%).

6. Conclusions and future work

The present research presents the development of an **optimised** sentiment analysis pipeline exploiting a **rich feature set** and investigated the effect of **complex normalisation** as a preprocessing step. Different feature types were exploited, including bags of words, word shape features, syntactic features and sentiment lexicon features. We performed a series of experiments that were evaluated against three baselines (i.e. random, majority class and word unigram), which were all improved at each step of our experimental setup. In a first step, we explored the effect of classifier optimisation by means of joint hyperparameter and feature selection using genetic algorithms. The results revealed that, when applied in its default parameter settings, the LIBSVM model consistently predicted the majority class. Optimisation of the model increased its performance from F_1= 21.30% to F_1= 63%, and eventually to F_1=75.29% when a rich set of features was exploited instead of bags of words.

Having in place an optimised and rich-feature-based sentiment analysis architecture, our focus was on investigating the usefulness of lexical normalisation as a preprocessing step to sentiment classification. Cross-validation experiments on the development data showed that the *norm* model did not outperform the *not norm* model, but experiments on an unseen and varied test set revealed that normalisation did benefit the classification performance. In fact, including normalisation as preprocessing not only improved the classification results for tweets, but also for LiveJournal blog posts,

7. http://www.cs.cmu.edu/ ark/TweetNLP.

a different type of user-generated content. This demonstrates that the *norm* model performs well on unseen data and that it is able to generalise to other genres of UGC than Twitter. For the noisier SMS genre, however, the *not norm* model performed better.

In a last step, we compared the performance of our optimised sentiment classifier (with and without normalisation as preprocessing) to that of the SemEval-2014 winning team and observed that our model performs best overall, and proves more robust to other genres of user-generated content than Twitter.

In future research, it will be interesting to explore "targeted" normalisation, which focuses on increasing the coverage of words that are relevant to the task (i.e. sentiment-bearing words). As was demonstrated in section 4.2.3, besides errors, normalisation also corrects character flooding and inconsistent capitalisation, which could hint at a specific sentiment. Hence, it will be interesting to investigate whether these forms of creative writing can be left unchanged and how this would affect the system performance. Exploring deep learning using neural networks for sentiment classification is another important direction for future work, as several top-ranked systems in the SemEval-2015 to -2017 competitions were based on deep learning. Since our approach involves an intensive process of feature engineering, it would be interesting to see whether competitive or even better results can be achieved when features are automatically deduced from the data using neural networks. Finally, we plan to port our sentiment analysis system to other languages, starting with Dutch.

Acknowledgements

We thank the researchers of TeamX for kindly sharing their SemEval-2014 submission with us, allowing us to compare the performance of our sentiment classifier to their approach.

The work presented in this paper was carried out in the framework of the AMiCA (Automatic Monitoring for Cyberspace Applications) IWT SBO-project 120007.

7. References

Agarwal A., Xie B., Vovsha I., Rambow O., Passonneau R., "Sentiment Analysis of Twitter Data", *Proceedings of the Workshop on Languages in Social Media (LSM 2011)*, ACL, p. 30-38, 2011.

Barbosa L., Feng J., "Robust Sentiment Detection on Twitter from Biased and Noisy Data", *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, ACL, p. 36-44, 2010.

Chandrashekar G., Sahin F., "A survey on feature selection methods", *Computers & Electrical Engineering*, vol. 40, n° 1, p. 16-28, 2014.

Chang C.-C., Lin C.-J., "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, n° 3, p. 27:1-27:27, 2011.

Dadvar M., Trieschnigg D., de Jong F., "Experts and Machines against Bullies: A Hybrid Approach to Detect Cyberbullies", *Advances in Artificial Intelligence – 27th Canadian Conference on Artificial Intelligence*, p. 275-281, 2014.

Daelemans W., Zavrel J., van der Sloot K., van den Bosch A., TiMBL: Tilburg Memory Based Learner, version 6.2, Reference Guide, Technical Report n° 09-01, ILK Research Group, 2009.

Davidov D., Tsur O., Rappoport A., "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys", *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, ACL, Stroudsburg, PA, USA, p. 241-249, 2010.

De Clercq O., Desmet B., Schulz S., Lefever E., Hoste V., "Normalization of Dutch user-generated content", *Proceedings of the 9th International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, INCOMA Ltd., p. 179-188, 2013.

De Clercq O., Schulz S., Desmet B., Hoste V., "Towards shared datasets for normalization research", *in* N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (eds), *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2014)*, ELRA, p. 1218-1223, 2014.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R., "Indexing by latent semantic analysis", *Journal of the American Society for Information Science*, vol. 41, n° 6, p. 391-407, 1990.

Desmet B., Finding the online cry for help: automatic text classification for suicide prevention, PhD thesis, Ghent University, 2014.

Desmet B., Hoste V., Verstraeten D., Verhasselt J., Gallop Documentation, Technical Report n° LT3 13-03, University of Ghent, 2013.

Eisenstein J., "What to do about bad language on the internet", *Proceedings of the North American Chapter of the ACL : Human Language Technologies (NAACL-HLT)*, ACL, p. 359-369, 2013.

Finn A., Kushmerick N., Smyth B., "Genre Classification and Domain Transfer for Information Filtering", *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research: Advances in Information Retrieval*, Springer-Verlag, London, UK, UK, p. 353-362, 2002.

Gimpel K., Schneider N., O'Connor B., Das D., Mills D., Eisenstein J., Heilman M., Yogatama D., Flanigan J., Smith N. A., "Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments", *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies (HLT 2011) – Volume 2*, ACL, p. 42-47, 2011.

Haddi E., Liu X., Shi Y., "The Role of Text Pre-processing in Sentiment Analysis", *Procedia Computer Science*, vol. 17, p. 26-32, 2013.

Han B., Cook P., Baldwin T., "Lexical Normalization for Social Media Text", *ACM Transactions on Intelligent Systems and Technology*, vol. 4, n° 1, p. 5:1-5:27, February, 2013.

Heafield K., "KenLM: Faster and Smaller Language Model Queries", *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT 2011)*, ACL, p. 187-197, 2011.

Hoste V., Optimization Issues in Machine Learning of Coreference Resolution, PhD thesis, Antwerp University, 2005.

Hsu C.-W., Chang C.-C., Lin C.-J., A Practical Guide to Support Vector Classification, Technical report, Department of Computer Science, National Taiwan University, 2003.

Hu M., Liu B., "Mining and summarizing customer reviews", *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD04*, ACM, p. 168-177, 2004.

Joshi M., Penstein-Rosé C., "Generalizing Dependency Features for Opinion Mining", *Proceedings of the ACL-IJCNLP 2009 Conference*, ACL, p. 313-316, 2009.

Kobus C., Yvon F., Damnati G., "Normalizing SMS: Are Two Metaphors Better Than One?", *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008) – Volume 1*, ACL, p. 441-448, 2008.

Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer C., Bojar O., Constantin A., Herbst E., "Moses: Open Source Toolkit for Statistical Machine Translation", *Proceedings of the ACL 2007 Demo and Poster Sessions*, p. 177-180, 2007.

Kohavi R., John G. H., "Wrappers for Feature Subset Selection", *Artificial Intelligence*, vol. 97, nº 1-2, p. 273-324, December, 1997.

Kökciyan N., Çelebi A., Özgür A., Üsküdarli S., "BOUNCE: Sentiment Classification in Twitter using Rich Feature Sets", *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, ACL, Atlanta, Georgia, p. 554-561, 2013.

Li C., Liu Y., "Normalization of text messages using character-and phone-based machine translation approaches", *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

Liu B., *Sentiment Analysis – Mining Opinions, Sentiments, and Emotions*, Cambridge University Press, 2015.

Liu X., Zhang S., Wei F., Zhou M., "Recognizing named entities in tweets", *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies (HLT 2011)*, ACL, p. 359-367, 2011.

Miura Y., Sakaki S., Hattori K., Ohkuma T., "TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data", *Proceedings of the International Workshop on Semantic Evaluation (SemEval 2014)*, ACL and Dublin City University, p. 628-632, August, 2014.

Moens M.-F., Li J., Chua T.-S., *Mining User Generated Content*, Chapman & Hall/CRC, 2014.

Mohammad S. M., "Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text", *in* H. Meiselman (ed.), *Emotion Measurement*, Elsevier, p. 201-226, 2016.

Mohammad S. M., Kiritchenko S., Zhu X., "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets", *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, ACL, Atlanta, Georgia, USA, p. 321-327, June, 2013.

Mohammad S. M., Turney P. D., "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon", *Proceedings of the NAACL HLT Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (CAAGET 2010)*, ACL, p. 26-34, 2010.

Mosquera A., Moreda P., "Improving Web 2.0 Opinion Mining Systems Using Text Normalisation Techniques", *Recent Advances in Natural Language Processing (RANLP) 2013*, 2013.

Nakov P., Ritter A., Rosenthal S., Sebastiani F., Stoyanov V., "SemEval 2016 Task 4: Sentiment Analysis in Twitter", *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, ACL, p. 1-18, 2016.

Nenkova A., "Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference", *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, AAAI Press, p. 1436-1441, 2005.

Nielsen F. A., "A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs", *in* M. Rowe, M. Stankovic, A.-S. Dadzie, M. Hardey (eds), *Proceedings of the ESWC2011 Workshop on "Making Sense of Microposts": Big things come in small packages*, vol. 718, CEUR-WS.org, p. 93-98, 2011.

Noreen E. W., *Computer-Intensive Methods for Testing Hypotheses: An Introduction*, Wiley-Interscience, April, 1989.

Özdemir C., Bergler S., "A comparative study of different sentiment lexica for sentiment analysis of tweets", *Proceedings of Recent Advances in Natural Language Processing (RANLP 2015)*, p. 488-496, 2015.

Pang B., Lee L., "Opinion Mining and Sentiment Analysis", *Foundations and Trends in Information Retrieval*, vol. 2, n° 1-2, p. 1-135, January, 2008.

Pang B., Lee L., Vaithyanathan S., "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques", *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, ACL, p. 79-86, 2002.

Pettersson E., Megyesi B., Tiedemann J., "An SMT approach to automatic annotation of historical text", *Proceedings of the workshop on computational historical linguistics at NODALIDA'13*, n° 87 in *NODALIDA'13*, Linköping University Electronic Press, Oslo, Norway, p. 54-69, 2013.

Raghunathan K., Krawczyk S., CS224N: Investigating SMS Text Normalization using Statistical Machine Translation, Technical report, Stanford University, 2009.

Ritter A., Clark S., Mausam, Etzioni O., "Named Entity Recognition in Tweets: An Experimental Study", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, ACL, p. 1524-1534, 2011.

Rosenthal S., Farra N., Nakov P., "SemEval 2017 Task 4: Sentiment Analysis in Twitter", *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, ACL, Vancouver, Canada, p. 502-518, 2017.

Rosenthal S., Nakov P., Kiritchenko S., Mohammad S., Ritter A., Stoyanov V., "SemEval 2015 Task 10: Sentiment Analysis in Twitter", *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, ACL, p. 451-463, June, 2015.

Rosenthal S., Ritter A., Nakov P., Stoyanov V., "SemEval 2014 Task 9: Sentiment Analysis in Twitter", *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, ACL and Dublin City University, Dublin, Ireland, p. 73-80, August, 2014.

Roy S., Dhar S., Bhattacharjee S., Das A., "Sentiment in Twitter Events", *Journal of the American Society for Information Science and Technology*, vol. 62, n° 2, p. 406-418, 2011.

Saif H., He Y., Alani H., "Alleviating Data Sparsity for Twitter Sentiment Analysis", *Proceedings of the #MSM Workshop*, vol. 838, CEUR-WS.org, p. 2-9, 2012.

Salton G., *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer*, Addison Wesley, 1989.

Schneider G., Pettersson E., Percillier M., "Comparing Rule-Based and SMT-Based Spelling Normalisation for English Historical Texts", *Proceedings of the NODALIDA'17 Workshop on Processing Historical Language*, n° 133 in *NODALIDA'17*, Linköping University Electronic Press, p. 40-46, 2017.

Severyn A., Moschitti A., "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification", *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, ACL, p. 464-469, 2015.

Sharma S., Srinivas P., Balabantaray R. C., "Text normalization of code mix and sentiment analysis", *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, p. 1468-1473, 2015.

Sokolova M., Lapalme G., "A systematic analysis of performance measures for classification tasks", *Information Processing & Management*, vol. 45, n° 4, p. 427-437, 2009.

Stone P. J., Dunphy D. C. D., Smith M. S., Ogilvie D. M., *The General Inquirer: A Computer Approach to Content Analysis*, The MIT Press, 1966.

Wiebe J. M., "Learning Subjective Adjectives from Corpora", *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, AAAI Press, p. 735-740, 2000.

Wiebe J. M., Wilson T., Bruce R., Bell M., Martin M., "Learning Subjective Language", *Computational Linguistics*, vol. 30, n° 3, p. 277-308, 2004.

Wilson T., Wiebe J., Hoffmann P., "Recognizing Contextual Polarity in Phrase-level Sentiment Analysis", *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005)*, ACL, p. 347-354, 2005.

Xue Z., Yin D., Davison B. D., "Normalizing Microtext", *Proceedings of the 5th AAAI Conference on Analyzing Microtext (AAAIWS 2011)*, AAAI Press, p. 74-79, 2011.

Yu L., Liu H., "Feature selection for high-dimensional data: a fast correlation-based filter solution", *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, p. 856-863, 2003.

Zhu X., Kiritchenko S., Mohammad S. M., "NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets", *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, ACL and Dublin City University, p. 443-447, 2014.