Expert Systems With Applications 110 (2018) 298–310

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis

Symeon Symeonidis*, Dimitrios Effrosynidis, Avi Arampatzis

Database & Information Retrieval Research Unit, Department of Electrical & Computer Engineering, Democritus University of Thrace, Xanthi 67100, Greece

ARTICLE INFO

Article history:
Received 28 December 2017
Revised 21 May 2018
Accepted 8 June 2018
Available online 15 June 2018

Keywords:
Sentiment analysis
Text pre-processing
Machine learning
Text classification
Ablation study
Combination study

ABSTRACT

Pre-processing is the first step in text classification, and choosing right pre-processing techniques can improve classification effectiveness. We experimentally compare 16 commonly used pre-processing techniques on two Twitter datasets for Sentiment Analysis, employing four popular machine learning algorithms, namely, Linear SVC, Bernoulli Naïve Bayes, Logistic Regression, and Convolutional Neural Networks. We evaluate the pre-processing techniques on their resulting classification accuracy and number of features they produce. We find that techniques like lemmatization, removing numbers, and replacing contractions, improve accuracy, while others like removing punctuation do not. Finally, in order to investigate interactions—desirable or otherwise—between the techniques when they are employed simultaneously in a pipeline fashion, an ablation and combination study is contacted. The results of ablation and combination show the significance of techniques such as replacing numbers and replacing repetitions of punctuation.

1. Introduction

In the last decade, Sentiment Analysis in microblogging has become a very popular research area. People share their daily life through messages on platforms such as Twitter, where posts of users involve various topics. Interesting approaches for classification methods in Sentiment Analysis are presented in many research papers (e.g. Agarwal, Xie, Vovsha, Rambow, & Passonneau, 2011; Mohammad, Zhu, Kiritchenko, & Martin, 2015), and the important role of pre-processing before and during the feature selection process is widely noted.

In this context, pre-processing is the procedure of cleansing and preparing texts that are going to be classified. It is a fact that unstructured texts on the Internet—in our case on Twitter—contain significant amounts of noise. By the term noise, we define data that do not contain any useful information for the analysis at hand, i.e. Sentiment Analysis.

According to Fayyad, Piatetsky-Shapiro, and Uthurusamy (2003), the total percentage of noise in datasets may reach 40%, a fact that causes confusion in machine learning algorithms. Twitter users are prone to spelling and typographical errors and to the use of abbreviations and slang. They may also (over- or mis-) use punctuation marks to emphasize their emotions, like for example many exclamation marks. Usually, it is not necessary to include all terms of the initial form of a text in the machine learning step, and some of them can be ignored, replaced, or merged with others. Thus, the need of cleansing and normalizing the data arises, as their quality is a key factor to the success of the machine learning that follows pre-processing.

The purpose of this study is to gather common pre-processing techniques from previous studies, add a few new ones that have not been used a lot by researchers, such as replacing contractions and replacing negations with antonyms, and examine their significance in feature selection by measuring their accuracy in sentiment classification and their resulting number of features.

In the end, based on the results obtained, the techniques that are more suitable for Twitter Sentiment Analysis and those that have to be avoided are suggested to future researchers. The present study is a comprehensive extension of our previous work (Effrosynidis, Symeonidis, & Arampatzis, 2017), and it also investigates the interactions among pre-processing methods via ablation and combination studies.

The rest of this paper is structured as follows. Section 2 reviews some of the related literature. In Section 3, the pre-processing techniques that will be compared are presented. Section 4 describes the datasets, the machine learning algorithms, and the evaluation methodology, while our results are presented and discussed in Section 5. Conclusions and directions for future research are summarized in Section 6.

* Corresponding author.
E-mail addresses: ssymeoni@ee.duth.gr (S. Symeonidis), deffrosy@ee.duth.gr (D. Effrosynidis), avi@ee.duth.gr (A. Arampatzis).

https://doi.org/10.1016/j.eswa.2018.06.022
0957-4174/

2. Related work

In Sentiment Analysis, especially on microblogging texts, the role of pre-processing techniques is significant as a part of text classification. Many research efforts have been made in order to demonstrate the difference between these techniques and their contribution to the final result of classification.

Singh and Kumari (2016) examine the effects of pre-processing on Twitter data for the fortification of sentiment classification. They focus on tweets which are full of symbols, abbreviations, folksonomy, and unidentified words. By removing URLs, hashtags, user mentions, punctuation, and stopwords, they recognize and accept the importance of slang words and spelling correction. In their experiments, an SVM classifier is employed.

Bao, Quan, Wang, and Ren (2014) studied the impact of pre-processing methods on Twitter sentiment classification, evaluating on Stanford Twitter Sentiment Dataset. The experimental results presented a positive effect on sentiment classification when using the pre-processing techniques of URLs features reservation, negation transformation, and repeated letters normalization, while stemming and lemmatization had a negative impact.

The role of pre-processing is also investigated by Haddi, Liu, and Shi (2013) on movie reviews. They use pre-processing techniques such as expansion of abbreviations, removal of non-alphabetic signs, stopword removal, negation handling with the addition of the prefix 'NOT_', and stemming. An SVM classifier is also employed and the authors correlate the number of features to its accuracy. It is shown that appropriate text pre-processing methods, including data transformation and filtering, can significantly enhance the classifier's performance.

Pre-processing techniques are also explored by Uysal and Günal (2014) for two languages on e-mails and news. They employ stopword removal, lowercase conversion, and stemming, and they evaluate with micro-averaged F_1 score using an SVM classifier. They conclude that there is no unique combination of pre-processing techniques that improves accuracy on any domain or language and the researchers should carefully analyse all possible combinations.

Zhao and Gui (2017), focused on effects of text pre-processing methods and used six pre-processing methods on five Twitter datasets with two feature models and four classifiers. The effectiveness of sentiment classification increased by the methods of expanding acronyms and replacing negations, and decreased by the methods of removing URLs, numbers, and stop words.

The Workshop on Noisy User-generated Text[1], which takes place annually since 2015, focuses on natural language processing applied to noisy user-generated text. In 2015, the workshop introduced a lexical normalization task, aiming at normalizing non-standard words in English Twitter messages to their canonical forms. In studies of Saloot, Idris, Shuib, Raj, and Aw (2015) and Yamada, Takeda, and Takefuji (2015), new approaches were presented to minimize the noise of Twitter messages, by using the maximum entropy model for normalizing Tweets and Entity Linking which is a method to detect entity mentions for text and resolve them to corresponding entries.

Other studies (Liao, Wang, Yu, Sato, & Cheng, 2017; Severyn & Moschitti, 2015; Tang, Wei, Yang et al., 2014) examined the deep learning approach for Sentiment Classification and used Convolutional Neural Networks and Word Embeddings in order to achieve better results than those obtained through traditional techniques. dos Santos and Gatti (2014) examined the creation of a network which took advantage of different levels of information to perform Sentiment Analysis, and used character-level, word-level, and sentence-level representations and features. The performance of manually-hand extracted features combining with automatically extracted embedding features by using deep learning techniques and integrating them with traditional approaches was examined by Araque, Corcuera-Platas, Sánchez-Rada, and Iglesias (2017). In the work of Tang, Wei, Qin, Liu, and Zhou (2014), the sentiment-specific word embedding features concatenated and annotated to become hand-crafted features for Twitter sentiment classification, and new features tested the latter to a deep neural network.

Despite the fact that many studies have examined the role of pre-processing, generally and specifically in Sentiment Analysis, none of them has gathered in a comparative study a large number of popular techniques as it is done in this work.

3. Common pre-processing techniques

As a first step in pre-processing, most (if not all) studies, e.g. Wang and Manning (2012), Symeonidis, Effrosynidis, Kordonis, and Arampatzis (2017), Pak and Paroubek (2010), Giachanou, Gonzalo, Mele, and Crestani (2017), apply tokenization. According to Balazs and Velásquez (2016) tokenization is "a task for separating the full text string into a list of separate words". Atkinson, Salas, and Figueroa (2015) defined tokenization as "a kind of lexical analysis that breaks a stream of text up into words, phrases, symbols, or other meaningful elements called tokens". At its core, the process of tokenization is a standard method for further Natural Language Processing (NLP) transformation in pre-processing.

The 16 pre-processing techniques we will experiment with are described below. The order that they should be applied is of major importance; we present them in the recommended order which enables combinations of them in the same pre-processing pipeline with as few adverse effects as possible. We briefly describe each technique, why it is applied, give an example, and mention related works that used it before.

3.1. Remove unicode strings and noise

Not all datasets are given clean. So, first of all, using some regular expressions we remove non-english characters and unicode strings like "\u002c" and "\x06" which were remnants of the crawling procedure that created the dataset. This technique is considered a baseline for our experiments.

3.2. Replacing URLs and user mentions

In Twitter texts, the majority of sentences contain a URL, a user mention, and/or a hashtag symbol. Their presence does not contain any sentiment and one approach is to replace them in pre-processing with tags as, e.g. Agarwal et al. (2011) do. In our work, we use the tags 'URL' and 'AT_USER' and removed the hashtag symbol. Some other thoughts could be to either remove only the punctuation signs in user mentions and keep the username or remove it completely (Bermingham & Smeaton, 2011; Khan, Bashir, & Qamar, 2014), but this case was not examined.

This technique is not universal and only applies to Twitter texts. So, it should/could be done before any other technique.

For example, the tweet

RT @BoomerLivingNow: Retirement: Don't Run Out of Money Before You Run Out of Time http://ow.ly/15RgI #finances #boomer #retirement

after this particular pre-processing step is transformed to:

RT AT_USER Retirement : Don't Run Out of Money Before You Run Out of Time URL finances boomer retirement

[1] http://noisy-text.github.io/.

Of course, before applying this technique (and most others presented next—unless otherwise stated), we remind that we have tokenized the inputs, as said above.

3.3. Replacing slang and abbreviations

Social media users usually write in an informal way and their texts contain a lot of slang and abbreviations. Slang is a type of language consisting of words and phrases that are regarded as very informal and are typically restricted to a particular context or group of people, while abbreviation is a shortened form of a word or phrase. These words and phrases, in order to be interpreted correctly, have to be replaced to impute their meaning. We manually constructed a lookup table consisting of 290 such words and phrases, and their replacements. Some examples are the phrases "ty", "qq" and "omg", which respectively mean and replace "thank you", "crying", and "oh my god".

For example, consider the following tweet:

@BillsAttention really? omg. At me was not bad. But I think that could will be better. haha and how you feel today?

It is transformed after this pre-processing step to:

@BillsAttention really? oh my god. At me was not bad. But I think that could will be better. haha and how you feel today?

Previous studies Kouloumpis, Wilson, and Moore (2011), Wu, Zhang, and Ostendorf (2010) are used this technique for Sentiment Analysis and classification.

3.4. Replacing contractions

One technique that can be used in pre-process is the replacement of contractions, i.e. strings like "won't" and "don't" will be replaced by "will not" and "do not", respectively.

For example, the text

Am I rdy 2 go back 2 Cali so soon after grandma passin away?? Don't know if a stop 2 Cali after Cabo is a good idea but I miss my fam. =(

will be transformed after this pre-processing step to:

Am I rdy 2 go back 2 Cali so soon after grandma passin away?? Do not know if a stop 2 Cali after Cabo is a good idea but I miss my fam. = (

If we do not replace contractions, the tokenization process would create the tokens "don" and "'t" (for the case of "don't"), with the second one not being particularly helpful as it will match with more than the other not's in texts. We also need the word "not" for a pre-processing technique that comes later, the one that replaces negations with antonyms. Boia, Faltings, Musat, and Pu (2013) and Sánchez-Mirabal et al. (2014) in previous studies, replace contractions with their full forms or by its relevant word.

3.5. Removing numbers

It is a common method to remove numbers from text, because they do not contain any sentiment. This step must come after the replacement of slang, because some slang words such as "gr8", which means "great", contain numbers.

For example, the tweet

@keishanichole @ashleybrittney Lol Keesh that's not nice! I will curse you 2 out later smh

is transformed after this pre-processing step to:

@keishanichole @ashleybrittney Lol Keesh that's not nice! I will curse you out later smh

Many do it, e.g. He, Lin, and Alani (2011), Zhao (2015), however, some researchers (Lin & He, 2009) argue that keeping the numbers may improve classification effectiveness.

3.6. Replacing repetitions of punctuation

We distinguish three punctuation signs, whose repetitions concern us. These are the exclamation, question, and stop marks. The use of these punctuation marks signals the existence of intense emotion. If we find more than one in a row, we replace it with a representative tag. For example, the token "???" will be replaced by "multiQuestionMark". This process must be done before removing punctuation shown later.

For example, the tweet

@AntiBarbieBitch girl u need to need to read just old dude. tryna roast me!!! but go through my page!

is transformed after this pre-processing step to:

@AntiBarbieBitch girl u need to need to read just old dude. tryna roast me multiExclamation but go through my page!

In study of Balahur (2013), this technique is used to normalize the language of tweets and generalize the vocabulary employed to represent sentiment.

3.7. Replacing negations with antonyms

It is an approach that has not been used by many researchers and is presented by Perkins (2010). We search in each sentence for the word "not" and then, we check if the next word has an antonym. For example, the phrase "not good" will be replaced with the word "bad". We replace both words with the unambiguous antonym, by using Wordnet (Miller, 1995).

For example, the text

i don't understand i really don't. this course feels wrong, hospital radio isn't right, and i'm not happy.

is transformed after this pre-processing step to:

i don't understand. i really don't. this course feels wrong, hospital radio isn't right, and i'm unhappy

This not commonly used technique examined by Chalil, Sendhilkumar, and Mahalakshmi (2015) and Medhat, Yousef, and Mohamed (2014), where every negation word was replaced if an unambiguous antonym is present in the WordNet.

3.8. Removing punctuation

For preprocessing of text, a classic technique in information retrieval and data mining is to removing punctuation. However, many times the presence of punctuation marks often denotes the existence of some sentiment. For example, an exclamation mark may mean an intense positive or negative sentiment. Hence, if we remove them we might decrease the accuracy of classification.

For example, the tweet

"Urgh i'm soo bored and tired. text. call, anything. hmm i must find things to multi task on now!"

is transformed after this pre-processing step to:

Urgh im soo bored and tired text call anything hmm i must find things to multi task on now

In many works, it is common to remove punctuation signs in pre-processing, see e.g. Lin and He (2009).

3.9. Handling capitalized words

Before we apply the popular technique of lowercasing, we handle capitalized words. Capitalized words may imply intense emotion, so we detect all the words that are longer than two characters with all of their characters capitalized.

For example, the tweet

@wendywave1 HAHAHAH that was worded weird. I'm reading while a candle is burning in my room

is transformed after this pre-processing step to:

@wendywave1 ALL_CAPS_HAHAHAH that was worded weird. I'm reading while a candle is burning in my room

As can be seen, we prefix them with "ALL_CAPS_" like in Prasad (2010), so they can be identified in machine learning.

3.10. Lowercasing

One of the most common pre-processing techniques is to lowercase all words. By doing so, the same words are merged and the dimensionality of the problem is reduced.

For example, the text

I spilled milk all up in my Macbook.

is transformed after this pre-processing step to:

i spill milk all up in my macbook.

The studies of dos Santos and Gatti (2014) and Zhang, Wu, and Lan (2015), lowercase all words to have clean token information.

3.11. Removing stopwords

Stopwords are function words with high frequencies of presence across all sentences. It is considered needless to analyze them, due to the fact that they do not contain much useful information for Sentiment Analysis. The set of these words is not completely predefined and it can be changed by removing or adding more to it, depending on the application.

For example, the tweet

Its time you changed direction! This is the answer! It'll blow your socks off! http://profusionXis4me.info

is transformed after this pre-processing step to:

time changed direction ! is answer ! blow socks off ! http://profusionXis4me.info

In our implementation, the standard stopwords provided by NLTK (Loper & Bird, 2002) were removed.

3.12. Replacing elongated words

Elongated is a word which it contains a character that is wrongly (but often purposely) repeated one or more times, e.g. "greeeeat". It is essential to replace words like this with their source words, so they can be merged. Otherwise, the classifier will treat them as different words, and probably the elongated ones will be ignored due to their low frequency of occurrence.

For example, the tweet

@NickWoodford93 goood:), alrite i supposee ¡33 up too muchh xx

is transformed after this pre-processing step to:

@NickWoodford93 good:), alrite i suppose ¡33 up too much xx

Detecting and replacing elongated words has been examined by researchers before, e.g. Mohammad, Kiritchenko, and Zhu (2013).

3.13. Spelling correction

It is very common in informal texts for users to make spelling errors that might make classification harder. Norvig's spelling corrector is employed in our current study.[2]

For example, the text

i just w0nder if u ever think of me..happy thoughts

is transformed after this pre-processing step to:

i just wonder if u ever think of me..happy thoughts

By using tools that automatically correct these spelling errors, it is possible to enhance classification effectiveness (Mullen & Malouf, 2006). While no corrector is perfect, they have some—usually high—accuracy of success.

3.14. Part-of-Speech (POS) tagging

It is the process by which each word is assigned a part-of-speech label. This label can be noun (NN), adverb (RB), verb (VB), or a more specialized part of speech like proper plural noun (NNPS), superlative adverb (RBS), 3rd person verb (VBZ), etc. The purpose of POS tagging in pre-processing is to exclude some parts of speech that do not contain any sentiment for the current application. Only nouns, verbs, and adverbs were kept in our study.

For example, the text

About to spend the next couple of hours of my life working for the man

is transformed after this pre-processing step to:

about spend next couple hour life work man

Some studies (Barbosa & Feng, 2010; Kouloumpis et al., 2011) use POS tags as pointers for sentiment tagging on opinion messages.

3.15. Lemmatization

Another method of merging many words to one is lemmatization. This method analyzes a word morphologically and removes its inflectional ending, producing its base form or lemma as it is found in a dictionary.

For example, the tweet

@ WCooksey I'm shocked, nay appalled, to hear that you are regularly communicating with your constituents. And caught on tape!

is transformed after this pre-processing step to:

@ WCooksey I'm shock, nay appal, to hear that you are regularly communicate with your constituent. And catch on tape!

Lemmatization is used by Guzman and Maalej (2014) to reduce the number of feature descriptors for user sentiment extraction.

3.16. Stemming

It is the process of removing the endings of the words in order to detect their root form or stem. By doing so, many words are merged and the dimensionality is reduced. It is a widely used method that generally yields good results; In our work, the Porter Stemmer (Porter, 1980) is used.

[2] http://norvig.com/spell-correct.html.

Table 1
Correspondence of pre-processing techniques.

Number	Pre-processing Technique
0	Basic (Remove Unicode strings and noise)
1	Other (Replace URLs and user mentions)
2	Replace Slang and Abbreviations
3	Replace Contractions
4	Remove Numbers
5	Replace Repetitions of Punctuation
6	Replace Negations with Antonyms
7	Remove Punctuation
8	Handling Capitalized Words
9	Lowercase
10	Remove Stopwords
11	Replace Elongated Words
12	Spelling Correction
13	Part of Speech Tagging
14	Lemmatizing
15	Stemming
16	Handling Negations

For example, the text

Saying good morning to everyone. Another gorgeous morning in Surrey. Wishing you all well.

is transformed after this pre-processing step to:

Say good morn to everyon. Anoth gorgeou morn in Surrey. Wish you all well

Mejova and Srinivasan (2011) refer the importance of stemming in polarity classification.

Lemmaziting and Stemming are mutually exclusive and they should not be used together.

3.17. Handling negations

When text analysis is performed in a word level, it is very challenging to handle negation. A method widely used by researchers is the detection of words that imply negation and the addition of the prefix 'NEG_' in every word after them until the first punctuation mark.

For example, the text

To the idiot outside, we didn't make the snowman right outside our window for no reason. Thanks for killing my first snowman jackass! :(

is transformed after this pre-processing step to:

To the idiot outside, we didn't make the snowman right outside our window for no NEG_ reason. Thanks for killing my first snowman jackass ! : (

This technique used by Narayanan, Arora, and Bhatia (2013) and contributed significantly to the accuracy of sentiment classifier.

Table 1summarizes and assigns numbers (for later use) to all the aforementioned techniques.

4. Experimental setup

Hitherto, several datasets for supervised Twitter Sentiment Analysis have been published. Each of them consists of tweets manually labeled by human annotators in one sentiment category. The most common categories are positive, negative, and neutral, but there are also some datasets which provide numeric labels that correspond to sentiment strengths.

Eight widely-used Twitter Sentiment Analysis datasets are presented by Saif, Fernández, He, and Alani (2013). We chose to examine the three-point classification problem, as the most popular approach, with the predefined classes of positive, negative, and neutral. For this task, two datasets were used, the first being the Sentiment Strength Twitter dataset and the second the SemEval dataset, both described next.

Table 2
Some statistics of the datasets.

	SS-Twitter	SemEval
Total Sentences	4242	65,854
Total Words	80,246	1,454,723
Average Words/Sentence	18.91	22.09
Total Unique Tokens	22,496	176,578
Total Emoticons	3467	34,979
Total Slangs	622	5815
Total Elongated Words	1543	17,355
Total Multi Exclamation Marks	325	2834
Total Multi Question Marks	152	750
Total Multi Stop Marks	1118	14,115
Total All Capitalized Words	2854	52,141

4.1. The sentiment strength twitter dataset

The Sentiment Strength Twitter or SS-Twitter dataset contains 4242 tweets and was developed by Thelwall, Buckley, and Paltoglou (2012) in order to evaluate SentiStrength,[3] a lexicon-based method for sentiment strength detection. The tweets are labelled with positive and negative strengths: a positive strength is a number between 1 ("not positive") and 5 ("extremely positive"), and a negative strength is a number between −1 ("not negative") and −5 ("extremely negative").

By re-annotating this dataset, a new one with three sentiment labels (positive, negative, neutral), suitable for our task was created. Hence, we apply two rules, as done in Saif et al. (2013). Firstly, we compute the positive to negative strength ratio of each tweet. If its absolute value is equal to 1, then we label the tweet as neutral. If the positive strength ratio is 1.5 times greater than the negative one, the tweet is considered positive. If it is the opposite we regard it as negative. After these transformations, the final dataset consists of 1252 positive, 1037 negative and 1953 neutral tweets. Some statistics related to the dataset are shown in Table 2.

4.2. The SemEval dataset

This dataset was constructed for the International Workshop on Semantic Evaluation (SemEval).[4] SemEval consists of many tasks and one of them is about three-point sentiment classification. Each tweet was manually annotated by Amazon Mechanical Turk workers or CrowdFlower users, depending on the year. This task has been run annually since 2013, Nakov et al. (2013), and every year more data are added. By collecting the datasets of all years (2013–2017), 65,854 tweets were gathered, i.e. 23,197 positive, 12,510 negative, and 30,147 neutral. Some statistics related to this dataset are also shown in Table 2.

4.3. Machine learning algorithms

Out of the many available supervised machine learning and deep learning algorithms, one algorithm for each of the four most-used categories was chosen. These categories are, the Generalized Linear Models (GLM), the Naïve Bayes (NB), the Support Vector Machines (SVM), and the Neural Networks (NN). From the GLM family we chose the Logistic Regression algorithm, from the NB we

[3] http://sentistrength.wlv.ac.uk.
[4] http://alt.qcri.org/semeval2017/.

chose the Bernoulli Naïve Bayes, from the SVMs we chose the Linear SVC algorithm, and for the NNs we chose Convolutional Neural Networks.

Logistic Regression (LR). It is a popular algorithm that belongs to the Generalized Linear Models methods—despite its name—and it is also known as Maximum Entropy. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function (Pedregosa et al., 2011). The previous studies of Lin, Mao, and Zeng (2017) and Wu, Huang, and Yuan (2017) used Logistic Regression for sentiment classification in microblogging.

Bernoulli Naïve Bayes (BNB). Naïve Bayes algorithms are the simplest probabilistic classification algorithms (John & Langley, 1995) that are widely used in Sentiment Analysis. They are based on the Bayes Theorem, which assumes that a complete independence of variables exists. The Bernoulli algorithm is an alternative of Naïve Bayes, where the weight of each term is equal to 1 if it exists in the sentence and 0 if not. Its difference from Boolean Naïve Bayes is that it takes into account terms that do not appear in the sentence. It is a fast algorithm that deals well with high dimensionality. For fitted Bernoulli Naïve Bayes on TF-IDF feature weighting scheme, we used TfidfTransformer from Pedregosa et al. (2011). Tripathy, Agrawal, and Rath (2016) and Ismail, Belkhouche, and Zaki (2018) used similar algorithms to address the Twitter sentiment analysis problem.

Linear SVC (LSVC). One of the most popular machine learning methods for classification of linear problems are SVMs (Cherkassky, 1997). They try to find a set of hyperplanes that separate the space into dimensions representing classes. These hyperplanes are chosen in a way which maximizes the distance from the nearest data point of each class. The Linear SVC is the simplest and fastest SVM algorithm assuming a linear separation between classes. Oliveira, Cortez, and Areal (2017), and Kang, Yoo, and Han (2012) used SVM classifiers for stock price forecasting with sentiment indicators and for sentiment analysis of restaurant reviews, respectively.

Convolutional Neural Networks (CNN). The first three algorithms are in fact linear classifiers. Naïve Bayes is a generative approach, whereas logistic regression and SVMs are discriminative approaches. Logistic Regression varies from SVMs in the fact that it provides a probabilistic interpretation for the results. Convolutional neural networks (CNNs) apply on local features, comprised of one or more convolutional layers and use them with convolving filters (Kim, 2014).The model takes as input the embedding of words sequentially as presented in a sentence, with the purpose to summarize the meaning of the sentence through layers of convolution. Then, the convolutional model is pooling each sentence in the final layer where a fixed length vectorial representation is reached (Hu, Lu, Li, & Chen, 2014). In the same way, Chen, Xu, He, and Wang (2017) and Liao et al. (2017) applied CNNs for sentiment sentence classification of twitter data.

4.4. Methods and measures

There are several ways to assess the features in a bag-of-words representation. In our study, Term Frequency – Inverse Document Frequency (TF.IDF) was chosen, which for a feature occurring in a document is given by

$$\text{TF.IDF} = f \log(N/df),$$

where f is the number of occurrences of the feature in the document, N is the total number of documents in the collection, and df is the number of documents that contain this feature (Na, Sui, Khoo, Chan, & Zhou, 2004).

For Neural Networks, the feature learning method was word embedding which is a dense, low-dimensional and real-valued vector for a word. Word representations are encoding in an embedding matrix by column vectors, and each column communicates with vocabulary's word-level embedding of the ith word (dos Santos & Gatti, 2014).

With a dataset as input, Python's NLTK (Loper & Bird, 2002) was used and a new file as output for each pre-processing technique was created. Depending on the technique, the final file had more or less total and unique tokens than the initial, as can be seen in Fig. 1 at the right vertical axis.

After the creation of the new pre-processed files, machine learning algorithms using scikit-learn (Pedregosa et al., 2011) were applied. For vectorization, the tf–idf transformation was used for the first three algorithms (Logistic Regression, Bernoulli Naïve Bayes, Linear SVC) and as features we employed uni-grams, so that we can identify whether and how the number of features has an impact on the classification results; their default parameters in Sklearn were used, as parameter optimization is beyond the interest of this study. For Convolutional Neural Networks, the well-known implementation of Kim (2014) was used. The CNN was trained on top of pre-trained word vectors from word2vec with little hyperparameter tuning and static vectors.

The metric that was used to evaluate the classification results is Accuracy, which is the ratio of correct classifications to all classifications. Accuracy is a good metric for balanced datasets, like in our case. Where there are orders of magnitude of class imbalances, other measures (such as F_β) may be more appropriate.

5. Results and discussion

After describing the selected pre-processing techniques, the machine learning algorithms, and evaluation measures, we are presenting and discussing the experimental results on the two datasets. First, we present the evaluation of each technique when used in isolation. Then, we perform an ablation study, i.e., apply the techniques all together but we switch one off in turns, in order to investigate cross-method interactions.

5.1. One technique at a time

Based on the overall results, we can discern 5 categories depending on the accuracy. These categories describe how the SS-Twitter and SemEval datasets reacted to the 16 pre-processing techniques for three-point Twitter Sentiment Analysis and are presented in Table 3.

Next we discuss the performance of each of the techniques in comparison to the basic baseline technique. These numbers are presented in Table 4. The green highlight shows the results that beat the baseline.

5.1.1. Replace URLs and user mentions

This technique has managed to get above the baseline, on two algorithms (Logistic Regression and Bernoulli Naïve Bayes) for both datasets. The measuring of user's influence on others by mentions and URLs can indicate how popular the user is and his affectiveness over a variety of topics (Cha, Haddadi, Benevenuto, & Gummadi, 2010).

5.1.2. Replace slang and abbreviations

The technique of replacing slang and abbreviations yields accuracy over the baseline for SS-Twitter dataset only on Bernoulli Naïve Bayes, and for SemEval dataset on Bernoulli Naïve Bayes and Linear SVC. The role of slang and abbreviations nowadays is

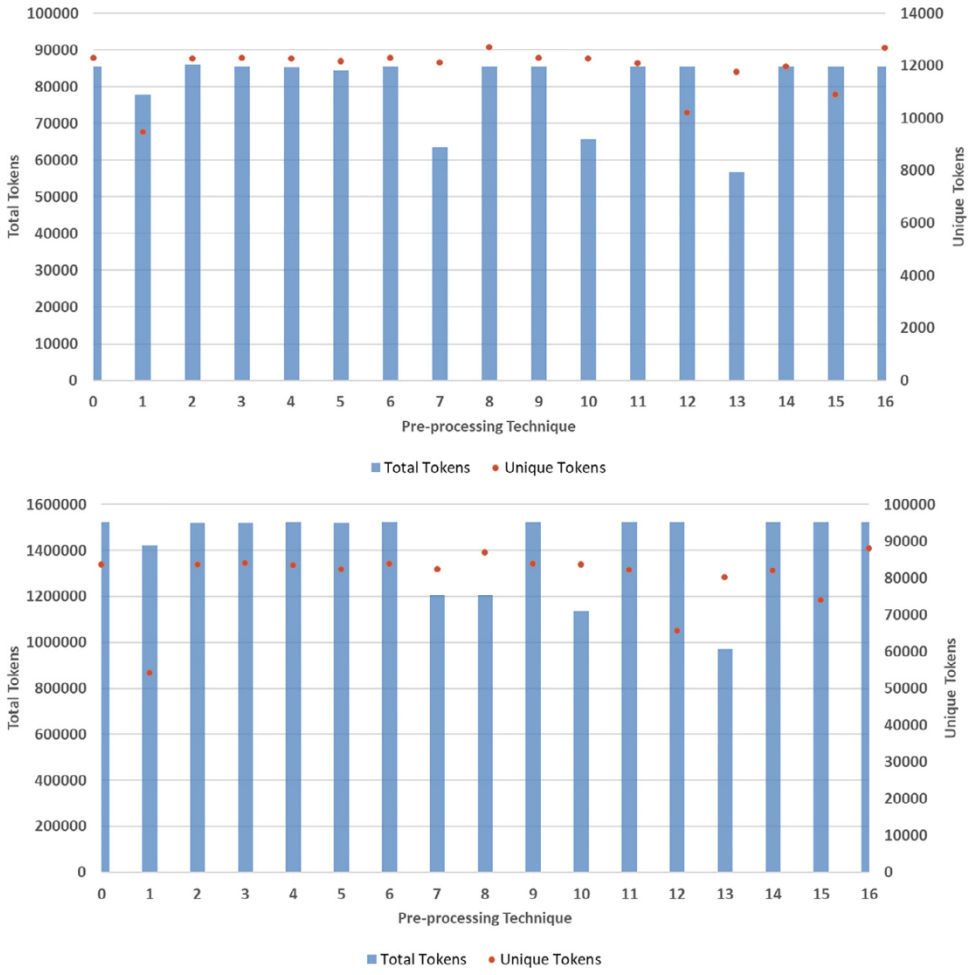

Fig. 1. Total and unique tokens per pre-processing technique in SS-Twitter (above) and SemEval (below) datasets.

Table 3
Accuracy performance categories for all pre-processing techniques on the SS-Twitter and SemEval dataset.

Performance	Description	SS-Twitter	SemEval
Best	High accuracy in all classifiers	3, 4, 5, 14, 15, 16	3, 4, 5, 10
High	High accuracy in most classifiers	1, 11	1, 2, 8, 13, 14
Poor	Low accuracy in most classifiers	2, 6, 8, 9, 12	6
Worst	Lowest accuracy in all classifiers	7, 10, 13	7, 9, 11, 12, 15, 16

Table 4
Accuracy percentage for each pre-processing technique on the SS-Twitter and SemEval dataset.

	SS-Twitter				SemEval			
Technique	LR	BNB	LSVC	CNN	LR	BNB	LSVC	CNN
0 - baseline	60.6	57.9	60.8	65.6	65.2	62.7	66.4	66.1
1	60.9	58.6	60.4	64.0	65.3	63.4	66.4	64.5
2	60.3	58.2	60.4	62.6	64.7	63.0	66.7	61.5
3	61.4	58.4	61.0	63.9	65.3	63.0	66.7	64.8
4	60.9	58.0	61.0	59.1	65.3	62.8	66.7	65.2
5	60.7	58.6	61.2	65.7	65.4	62.9	66.9	65.4
6	60.8	57.9	60.7	63.0	65.2	62.6	66.2	66.5
7	57.8	55.8	57.3	55.5	65.2	62.2	65.4	62.3
8	60.2	57.7	60.3	66.1	65.2	62.8	66.4	67.3
9	60.6	58.0	60.7	65.2	64.5	61.5	64.9	63.4
10	59.9	56.4	59.6	64.4	65.5	62.9	66.6	65.6
11	60.6	58.4	61.4	64.8	64.1	62.0	64.3	64.9
12	59.3	58.5	58.6	60.1	65.1	61.5	64.1	61.7
13	59.5	57.3	57.9	62.6	65.4	63.5	65.7	63.6
14	60.9	58.7	61.0	64.1	65.4	63.0	65.7	63.9
15	61.7	60.6	60.9	63.6	65.1	62.6	65.5	62.2
16	61.0	58.8	61.0	66.9	65.1	62.0	65.9	64.9

of major importance on short text messages due to the fact that it communicates the information quicker than the whole word (Haas, Takayoshi, Carr, Hudson, & Pollock, 2011). However, the origin of users and the topic of tweets are the main factors of occurrence of these acronyms.

5.1.3. Replace contractions

One of the most successful preprocessing techniques is replacing contractions, as on both Datasets the accuracy score is over the baseline for the three classic algorithms. Twitter users commonly use contractions in tweets, and sentiment lexicons exempt these phrases (Chalil et al., 2015).

5.1.4. Remove numbers

Another successful technique is removing numbers. This method managed to get over the baseline accuracy, for three algorithms on both Datasets. Numbers do not involve any sentiment, produce noise, and when taken out in preprocessing, tweet content is refined.

5.1.5. Replace repetitions of punctuation

A winning technique with good results is replacing repetitions of punctuation. On SS-Twitter dataset for whole algorithms and on SemEval dataset for the first three algorithms, this technique provides us with results over the baseline. According to Thelwall et al. (2012), replacing multiple punctuation marks, composes a forceful approach for the machine learning on microblogging text. The positive results of this technique is explained by the fact that both datasets are from Twitter and repeated punctuation on this source data specify sentiment connection.

5.1.6. Replace negations with antonyms

When employing this technique, only one algorithm (Logistic Regression) for SS-Twitter dataset and one (Convolutional Neural Networks) for SemEval dataset have managed to beat the baseline. The study of Xia et al. (2015) examines the polarity shift and shows that many machine learning algorithms fail when replacing negations with absolute antonyms. Many times the sense of the unambiguous antonym is not the correct one. This seems to be the reason for the failure of this technique.

5.1.7. Remove punctuation

A very typical technique for IR applications, which however does not provide us with good results on preprocessing for Sentiment Analysis, is removing punctuation. In both datasets and for all algorithms, the results were below the baseline of accuracy. Many microblogging users, especially on Twitter, do not often use punctuation marks due to the limited number of characters allowed, or they use emojis and multiple punctuation marks in order to express their feelings.

5.1.8. Handling capitalized words

Another characteristic technique, which has no impact on classic algorithms but only in Convolutional Neural Networks, is handling capitalized words. For both datasets, the accuracy was over the baseline for Neural Networks approach and for SemEval dataset was only on Bernoulli Naïve Bayes algorithm. The immaterial differences that this technique yields, can be attributed to the small number of capitalized words in the datasets (see Table 2). Moreover, capitalization and de-capitalization of words is essential in others domains, like when identifying news titles and determining the part-of-speech of each word appearing in the title, and not in domains as microblogging messages (Chaumartin, 2007).

5.1.9. Lowercase

One technique which did not manage to exceed the baseline accuracy, for both datasets and all algorithms, was lowercasing. Results of lowercase conversion contrary to the findings of Uysal and Günal (2014) where this technique proved to be helpful. The choice of domains can explain the difference on experimental findings; they used e-mail and news, while we used microblogging text.

5.1.10. Remove stopwords

The technique of removing stopwords yielded ambiguous results. For the SS-Twitter dataset, none of the algorithms was over the baseline accuracy but for the SemEval dataset, on three classic algorithms, the results were satisfactory.

The reasons for failure are: first, stopwords like 'I', 'me', 'you', present and are associated with expressions of sentiment (Thelwall et al., 2012), second, the domain of Tweets for each dataset, and third the vocabulary and the age of users. According to Haas et al. (2011) young people tend to use more and more short text with slangs and many stopwords to express their feelings about themselves.

5.1.11. Replace elongated words

For SS-Twitter dataset two algorithms (Bernoulli Naïve Bayes and Linear SVC) passed the baseline accuracy but for SemEval dataset none of the algorithms was over the baseline. Table 2 displays the small number of elongated words compared to the total number of words for the SemEval dataset, and the fair amount of elongated words compared to the total number of words for the SS-Twitter dataset. So, a small number of elongated words according to the total amount of words is giving better sentiment classification results.

5.1.12. Spelling correction

Spelling correction is also a technique which did not achieve good results. The only positive result of this technique stems from SS-Twitter dataset and Bernoulli Naïve Bayes algorithm. This technique fails because users benefit from spellchecker application and accidental mistakes during typing are few. Also, the spelling corrector that was used in our experiments is simple and with a better one the results may improve.

5.1.13. Part of speech tagging

The common NLP technique of POS tagging has managed to pass the baseline only for two algorithms (Logistic Regression and Bernoulli Naïve Bayes) and just for the SemEval dataset. As reported in a previous study (Kouloumpis et al., 2011), POS tags are losing the classification performance, and are limited helpful on Sentiment Analysis on microblogging data.

5.1.14. Lemmatizing

Lemmatizing, a frequently used technique, has managed to pass the baseline results for both datasets and especially for three classic algorithms for SS-Twitter dataset and for Logistic Regression and Bernoulli Naïve Bayes for SemEval dataset. According to Shotaroo, Takamura, and Okumura (2005), in a large dataset, lemmatizing words ignores semantic information and produces damage in the conjugated forms. Thus, this technique works for both microblogging and small—regarding the number of Tweets—datasets.

5.1.15. Stemming

The most popular technique of stemming has managed to exceed the baseline results only for the SS-Twitter dataset and for all algorithms except for CNN. However, for the SemEval dataset, it did not achieve the expected results. From our point of view, this failure in the CNN results can be demonstrated by the fact that stem

word shows the exact word and not a different one, thus resulting in a model that is learning on same representations (Maas et al., 2011). Furthermore, according to Annett and Kondrak (2008), stemming improves and assists in the loss of information but does not support enough the procedure of classification.

5.1.16. Handling negations

Finally, the technique of handling negations provided us with promising results only for the SS-Twitter dataset for all the algorithms. In the SemEval dataset, none of the algorithms managed to go beyond the baseline. The percentage of sentences that were influenced by this technique was 6% for SS-Twitter and only 1% for SemEval, and this might explain the above results. According to Wilson, Wiebe, and Hoffmann (2005), the phrases that include negative words do not change the polarity of text but boost it.

5.1.17. Summary of individual techniques

The previous subsections have shown the effectiveness of the different techniques applying to the datasets. According to Tables 3 and 4, five techniques overcome the baseline accuracy at least on two or three algorithms on both datasets. The rest of the techniques beat the baseline result for two algorithms at only one dataset, and a few did not manage to achieve any promising results.

According to Fig. 1, the number of Total tokens and Unique tokens changes according to the dataset. For the SS-Twitter dataset, the technique of replacing slang and abbreviations (number 2) has the biggest number of Total tokens while the technique of Part of Speech Tagging (number 13) the fewest. Concerning Unique tokens, the technique of handling capitalized words (number 8) is ranked first while at the last place lays technique number 1 (replace URLs and user mentions). For the SemEval dataset, the technique of handling negations (number 16) has the largest number of Total tokens, and Part of Speech Tagging (number 13) the smaller number. Furthermore, the technique with the biggest number of Unique tokens is handling negations (number 16) and technique of spelling correction with the fewest number.

In conclusion, if we want to reach the highest accuracy on three-point sentiment classification, we should use the combination of replacing URLs and user mentions, replacing contractions, removing numbers, replacing repetitions of punctuation, and lemmatizing. Moreover, if we aim at the best trade-off between the number of features and accuracy, we prefer the techniques of replacing URLs and user mentions, removing numbers, and replacing repetitions of punctuation.

5.2. Ablation and combination study

In the past, many studies have done an ablation analysis (Günther & Furrer, 2013; Hassan, Abbasi, & Zeng, 2013; Mohammad et al., 2013; Wilson et al., 2005). As stated in the above mentioned experimental results, in Table 4, where each technique of preprocessing was employed separately, it is possible for us to identify and choose the best five techniques when used individually taking into consideration that the remaining techniques brought worse results. Moreover, all possible combinations of 16 pre-processing techniques are $2^{16} - 1 = 65535$, a prohibitive number of cases for an exhaustive evaluation; thus our experimental approach is designed with a modest computational cost in mind. However, it is common in the community to use at least two or more techniques together during pre-processing.

In order for us to identify and examine interactions between techniques, an ablation study for the total number of techniques is performed first, which is the process of removing one technique at a time. Secondly, a combination study, in accordance with the previous results, is carried out in order to investigate the interaction between the best five techniques by removing one of them each time. This specific experimental route is followed due to the enormous amount of all possible combinations. In the following subsections, the results of ablation and combination study are discussed.

5.2.1. Technique ablation

In this section, the results when a technique is missing are discussed. We only focus to the results with a significant increase or decrease in classification accuracy when a technique is missing. As significant differences we arbitrarily defined those changes being larger than 2%. Fig. 2 presents the % change in accuracy when each of the techniques is missing.

On the one side, as we can see from the SS-Twitter dataset, when the techniques of replacing URLs and user mentions (number 1), replacing repetitions of punctuation (number 5), lowercasing (number 9), and stemming (number 15) are missing, the accuracy of the four classifiers is decreasing, a fact that displays the importance of these techniques. For the SemEval dataset, only when the technique of lowercasing (number 9) is excluded, the accuracy of classifiers drops, but not significantly.

On the other side, we observe the significant increase in accuracy of classifiers when a technique is missing. This points to a poor interaction between these techniques and the rest. For both datasets, when POS Tagging is missing the classification results are better. The same increase is observed for the removing stopwords technique. In general, in the case of both datasets and concerning all techniques, when lack of technique in the classification with Neural Networks is observed, the accuracy is increased.

To sum up, for the Logistic Regression algorithm on the one side, the most important techniques were replacing repetitions of punctuation and stemming. On the other side, the ablation of the techniques replace contractions, remove stopwords and part of speech tagging result in an increase in the classifier's accuracy. For Bernoulli Naïve Bayes, when removing stopwords and part of speech tagging are missing, the classification for Sentiment Analysis is working better, but if stemming is missing from preprocessing, then the accuracy is falling. When we are using Linear SVC, if the techniques of replacing contractions, removing stopwords, spelling correction, and part of speech tagging are missing from pre-processing, we have better results for classification. Essential techniques for this algorithm are replaced repetitions of punctuation, lowercase and stemming. For Neural Networks, training the model with more data, without much pre-processing, is a better way to have significant experimental results.

To conclude, in case we want to employ the rest of techniques for pre-processing, a possible scenario according to the experimental results of previous studies, it is necessary for us to replace URLs and user mentions (number 1), replace repetitions of punctuation (number 5), lowercase (number 9), and stemming (number 15). Furthermore, if POS Tagging and removing stopwords are missing from pre-processing, we can possibly expect better results on sentiment classification, as they interact poorly with the rest of techniques.

5.2.2. Technique combination

Following the experiments for the 16 preprocessing techniques and the ablation analysis of them, we attempted to combine the best five techniques from Table 3, and developed a combination ablation study, for the rest of classifiers in both datasets. We chose the techniques of replacing URLs and user mentions (number 1), replacing contractions (number 3), removing numbers (number 4), replacing repetitions of punctuation (number 5), and lemmatization (number 14).

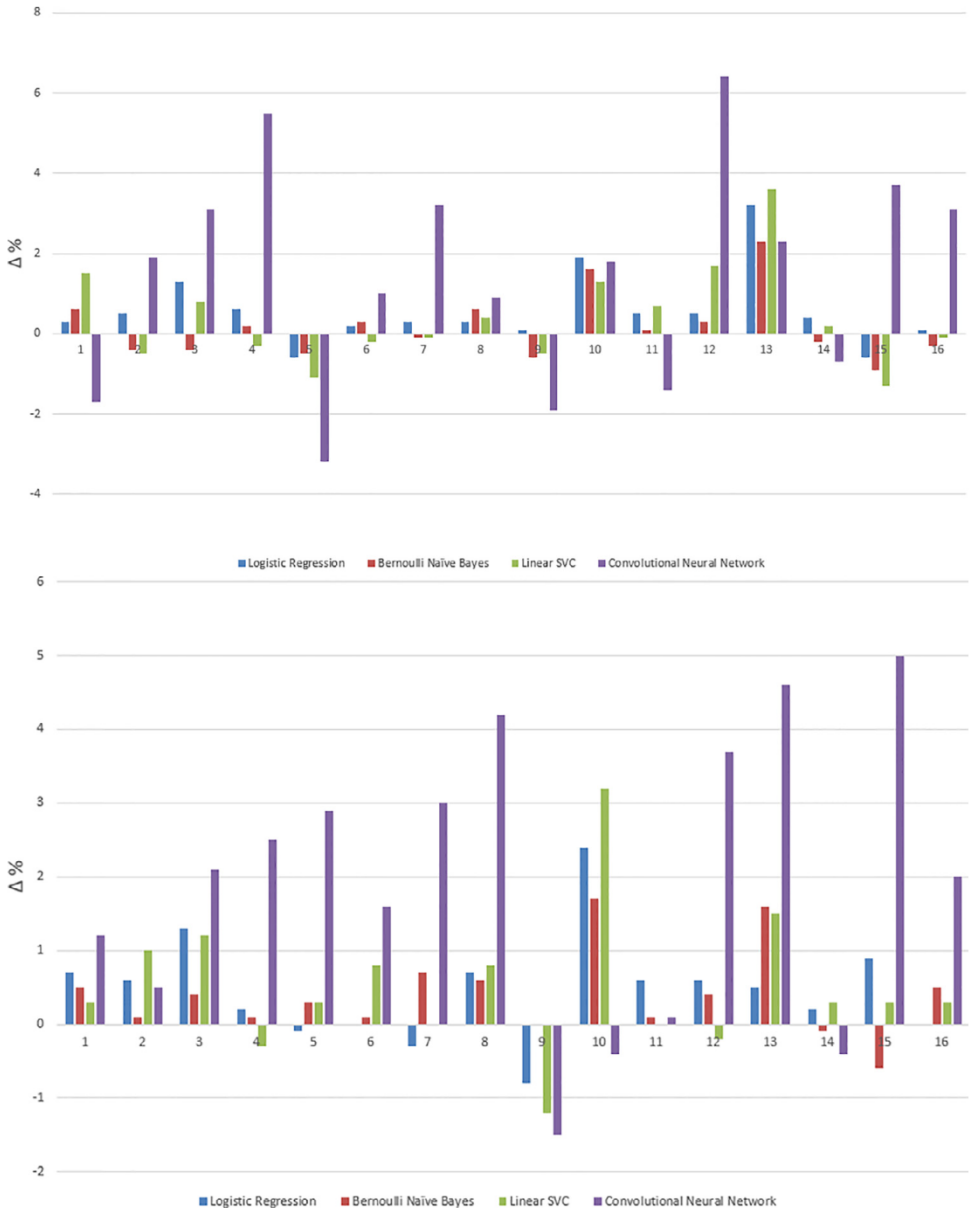

Fig. 2. Relative change in accuracy ablating a single pre-processing technique from the model that uses all of them in SS-Twitter (above) and SemEval (below) datasets.

Table 5
Accuracy percentage for each combination of pre-processing techniques on SS-Twitter and SemEval dataset.

	SS-Twitter				SemEval			
Combination	LR	BNB	LSVC	CNN	LR	BNB	LSVC	CNN
1-3-4-5-14	62.1	59.6	60.6	59.4	65.7	64	66.1	61.3
1-3-4-5	61.9	59.8	60.3	62.8	65.7	63.3	67.2	57.9
1-3-4-14	62.1	59.4	61.1	55.8	65.4	64	65.7	65.6
1-3-5-14	61.7	59.5	60.5	61.8	65.6	63.2	66.4	64.4
1-4-5-14	61.7	59.6	61	57.7	65.7	63.9	66.3	61.7
3-4-5-14	61.4	60	60.3	55.9	65.7	64	66.1	63.4

Table 5 presents technique ablation of combinations per dataset. The red highlight in cells of tables shows the results that reduce the accuracy of combination and do not pass the baseline results, while the green ones display results that increase the accuracy. The accuracy of every combination is compared to the accuracy of baseline results and only the results with a significant increase or decrease (larger than 2%) in accuracy of classifier when a technique is missing are mentioned. Fig. 3 presents the difference per combination for both datasets.

It was identified that in the first three classic algorithms of the combination study, there was not any significant increase or decrease on the accuracy of classifiers. On the contrary for CNN, depending on the dataset, the results differ. For the combination study when removing the technique of lemmatizing (number 14), we noticed an increase of accuracy on CNN by 3.35% on the SS-

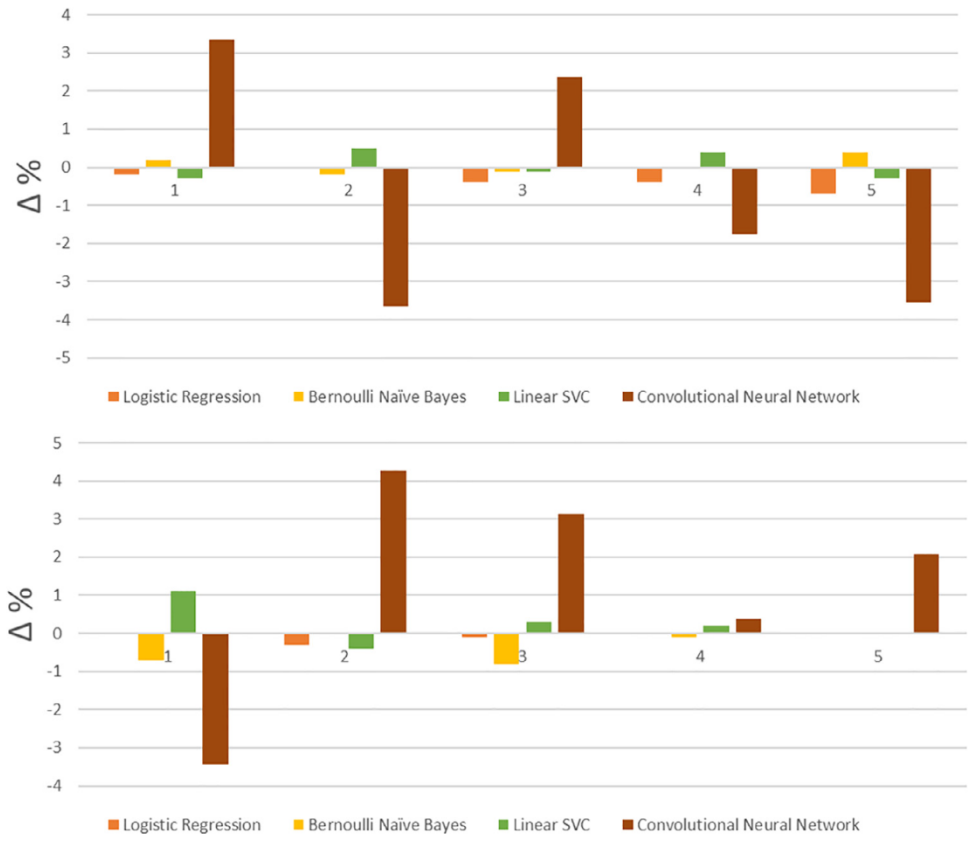

Fig. 3. Relative change in accuracy ablating a combination of top preprocessing techniques from the model that uses all of them in SS-Twitter (above) and SemEval (below) datasets.

Twitter dataset but on SemEval dataset, the decrease of CNN was 3.44%. In the same way, the removed technique number 5 (replace repetitions of punctuations) reduces the accuracy for CNN by 3.65% on the SS-Twitter dataset and increases it by 4.28% for CNN on SemEval dataset. Contrary to all the above, the missing technique number 4 (remove numbers) increased the accuracy for CNN by 2.35% on the SS-Twitter dataset, and by 3.13% for CNN on the SemEval dataset. Finally, when removing the technique of replacing URLs and user mentions, we noticed a reduction of accuracy on CNN by 3.55% on the SS-Twitter dataset and for the SemEval dataset, the increase of CNN was 2.07%.

For sure, the removing numbers technique seems to interact poorly with the other four techniques and can be missing from the combination. On the SS-Twitter dataset, lemmatizing and removing numbers techniques can be missing in the combination, but replacing repetitions of punctuation and replacing URLs and user mentions are essential for good accuracy. On the SemEval dataset, the most important technique is replacing URLs and user mentions and must be in the combination. For the rest of techniques, the absence of them yields better sentiment classification results.

Summarizing the results of the combination of the best five pre-processing techniques: It is beneficial to use the above preprocessing combination with classic machine learning algorithms for Sentiment Analysis on microblogging data. In the case of using Neural Networks, the combination of the five above-mentioned techniques is also suitable, but we can remove the technique of removing numbers to have better accuracy on sentiment classification.

6. Conclusion and future work

Sentiment analysis in microblogging platforms is an essential tool for research and business applications. The analysis of human sentiment and the understanding of human writings by machine learning processes help us to extract useful conclusions about human behavior. Pre-processing is the first step in text Sentiment Analysis, and the use of appropriate techniques can improve classification effectiveness.

We examined a significant number of pre-processing techniques, which have not been evaluated in a comparative study in the past, and tested them in two datasets. Each technique was evaluated in four representative machine learning algorithms on accuracy. Moreover, we distinguished some performance categories based on the results and counted the resulting number of features for each technique. Finally, an ablation study was performed for all, as well as for the high performance techniques, in order to determine their interactions.

Our experiments show that on Twitter Sentiment Analysis some techniques provide better results in classification for both of the datasets used, while others decrease the accuracy. The recommended techniques are lemmatization, replacing repetitions of punctuation, replacing contractions, and removing numbers. The non-recommended techniques include removing punctuation, marking up capitalized words handling capitalized words, replacing slang, replacing negations with antonyms, and spelling correction.

Depending on the classifier, the results vary, and if we combine these techniques we may get different results. A winning combination if someone wants to preprocess text for a classic machine learning Sentiment Analysis is: replace URLs and user mentions, replace Contractions, remove Numbers, replace repetitions of punc-

tuation, and lemmatizing. If we choose a Neural Network approach the above combination, without the technique of removing numbers, is the best. In future studies, we will test these techniques on datasets from different domains such as news articles and product or movie reviews and add new features. Finally, it would be interesting to test these techniques on a supervised dataset with emotion labels like anger, disgust, fear, happiness, sadness, and surprise, and see how they differ from the classic three-point classification problem that was examined in this work.

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*. In *LSM '11* (pp. 30–38). Stroudsburg, PA, USA: Association for Computational Linguistics.

Annett, M., & Kondrak, G. (2008). A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Advances in artificial intelligence, 21st conference of the Canadian society for computational studies of intelligence, Canadian AI 2008, Windsor, Canada, May 28–30, 2008, proceedings* (pp. 25–35). doi:10.1007/978-3-540-68825-9_3.

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications, 77*, 236–246. doi:10.1016/j.eswa.2017.02.002.

Atkinson, J., Salas, G., & Figueroa, A. (2015). Improving opinion retrieval in social media by combining features-based coreferencing and memory-based learning. *Information Sciences, 299*, 20–31. doi:10.1016/j.ins.2014.12.021.

Balahur, A. (2013). Sentiment analysis in social media texts. In A. Balahur, E. V. der Goot, & A. Montoyo (Eds.), *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis, wassa@naacl-hlt 2013, 14 June 2013, Atlanta, Georgia, USA* (pp. 120–128). The Association for Computer Linguistics.

Balazs, J. A., & Velásquez, J. D. (2016). Opinion mining and information fusion: A survey. *Information Fusion, 27*, 95–110. doi:10.1016/j.inffus.2015.06.002.

Bao, Y., Quan, C., Wang, L., & Ren, F. (2014). The role of pre-processing in twitter sentiment analysis. In *Intelligent computing methodologies - 10th international conference, ICIC 2014, Taiyuan, China, August 3–6, 2014. proceedings*. In *Lecture Notes in Computer Science: 8589* (pp. 615–624). Springer. doi:10.1007/978-3-319-09339-0_62.

Barbosa, L., & Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data. In C. Huang, & D. Jurafsky (Eds.), *COLING 2010, 23rd international conference on computational linguistics, posters volume, 23–27 august 2010, beijing, china* (pp. 36–44). Chinese Information Processing Society of China.

Bermingham, A., & Smeaton, A. (2011). On using twitter to monitor political sentiment and predict election results. In *Proceedings of the workshop on sentiment analysis where ai meets psychology (saaip 2011)* (pp. 2–10).

Boia, M., Faltings, B., Musat, C. C., & Pu, P. (2013). A :) is worth a thousand words: How people attach sentiment to emoticons and words in tweets. In *International conference on social computing, socialcom 2013, socialcom/passat/bigdata/econcom/biomedcom 2013, Washington, DC, USA, 8–14 september, 2013* (pp. 345–350). IEEE Computer Society. doi:10.1109/SocialCom.2013.54.

Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, P. K. (2010). Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the fourth international conference on weblogs and social media, ICWSM 2010, Washington, DC, USA, may 23–26, 2010* (p. 30).

Chalil, R. P., Sendhilkumar, S., & Mahalakshmi, G. S. (2015). Twitter sentiment analysis for large-scale data: An unsupervised approach. *Cognitive Computation, 7*(2), 254–262. doi:10.1007/s12559-014-9310-z.

Chaumartin, F. (2007). UPAR7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th international workshop on semantic evaluations, semeval@acl 2007, Prague, Czech Republic, June 23–24, 2007* (pp. 422–425).

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using bilstm-crf and CNN. *Expert Systems with Applications, 72*, 221–230. doi:10.1016/j.eswa.2016.10.065.

Cherkassky, V. (1997). The nature of statistical learning theory. *IEEE Transactions on Neural Networks, 8*(6), 1564. doi:10.1109/TNN.1997.641482.

Effrosynidis, D., Symeonidis, S., & Arampatzis, A. (2017). A comparison of pre-processing techniques for twitter sentiment analysis. In *Research and advanced technology for digital libraries - 21st international conference on theory and practice of digital libraries, TPDL 2017, Thessaloniki, Greece, September 18–21, 2017, proceedings* (pp. 394–406). doi:10.1007/978-3-319-67008-9_31.

Fayyad, U. M., Piatetsky-Shapiro, G., & Uthurusamy, R. (2003). Summary from the KDD-03 panel: Data mining: The next 10 years. *SIGKDD Explorations, 5*(2), 191–196. doi:10.1145/980972.981004.

Giachanou, A., Gonzalo, J., Mele, I., & Crestani, F. (2017). Sentiment propagation for predicting reputation polarity. In *Advances in information retrieval - 39th European conference on IR research, ECIR 2017, Aberdeen, UK, April 8–13, 2017, proceedings* (pp. 226–238). doi:10.1007/978-3-319-56608-5_18.

Günther, T., & Furrer, L. (2013). GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Proceedings of the 7th international workshop on semantic evaluation, semeval@naacl-hlt 2013, Atlanta, Georgia, USA, june 14–15, 2013* (pp. 328–332).

Guzman, E., & Maalej, W. (2014). How do users like this feature? A fine grained sentiment analysis of app reviews. In T. Gorschek, & R. R. Lutz (Eds.), *IEEE 22nd international requirements engineering conference, RE 2014, Karlskrona, Sweden, August 25–29, 2014* (pp. 153–162). IEEE Computer Society. doi:10.1109/RE.2014.6912257.

Haas, C., Takayoshi, P., Carr, B., Hudson, K., & Pollock, R. (2011). Young people's everyday literacies: The language features of instant messaging. http://www.jstor.org/stable/23050580. 10.2307/23050580

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. In *Proceedings of the first international conference on information technology and quantitative management, ITQM 2013, Dushu lake hotel, Sushou, China, 16–18 May, 2013* (pp. 26–32). doi:10.1016/j.procs.2013.05.005.

Hassan, A., Abbasi, A., & Zeng, D. (2013). Twitter sentiment analysis: A bootstrap ensemble framework. In *International conference on social computing, socialcom 2013, socialcom/passat/bigdata/econcom/biomedcom 2013, Washington, DC, USA, 8–14 September, 2013* (pp. 357–364). doi:10.1109/SocialCom.2013.56.

He, Y., Lin, C., & Alani, H. (2011). Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In D. Lin, Y. Matsumoto, & R. Mihalcea (Eds.), *The 49th annual meeting of the association for computational linguistics: Human language technologies, proceedings of the conference, 19–24 June, 2011, Portland, Oregon, USA* (pp. 123–131). The Association for Computer Linguistics.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27: Annual conference on neural information processing systems 2014, December 8–13 2014, Montreal, Quebec, Canada* (pp. 2042–2050).

Ismail, H. M., Belkhouche, B., & Zaki, N. (2018). Semantic twitter sentiment analysis based on a fuzzy thesaurus. *Soft Computing*. doi:10.1007/s00500-017-2994-8.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *UAI '95: Proceedings of the eleventh annual conference on uncertainty in artificial intelligence, Montreal, Quebec, Canada, August 18–20, 1995* (pp. 338–345).

Kang, H., Yoo, S. J., & Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications, 39*(5), 6000–6010. doi:10.1016/j.eswa.2011.11.107.

Khan, F. H., Bashir, S., & Qamar, U. (2014). TOM: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems, 57*, 245–257. doi:10.1016/j.dss.2013.09.004.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a special interest group of the ACL* (pp. 1746–1751).

Kouloumpis, E., Wilson, T., & Moore, J. D. (2011). Twitter sentiment analysis: The good the bad and the omg!. In *Proceedings of the fifth international conference on weblogs and social media, Barcelona, Catalonia, Spain, July 17–21, 2011* (pp. 538–541).

Liao, S., Wang, J., Yu, R., Sato, K., & Cheng, Z. (2017). CNN for situations understanding based on sentiment analysis of twitter data. *Procedia Computer Science, 111*(00), 376–381. doi:10.1016/j.procs.2017.06.037.

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on information and knowledge management, CIKM 2009, Hong Kong, China, November 2–6, 2009* (pp. 375–384). doi:10.1145/1645953.1646003.

Lin, J., Mao, W., & Zeng, D. D. (2017). Personality-based refinement for sentiment classification in microblog. *Knowl.-Based Syst., 132*, 204–214. doi:10.1016/j.knosys.2017.06.031.

Loper, E., & Bird, S. (2002). Nltk: The natural language toolkit. In *Proceedings of the acl-02 workshop on effective tools and methodologies for teaching natural language processing and computational linguistics - volume 1*. In *ETMTNLP '02* (pp. 63–70). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1118108.1118117.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *The 49th annual meeting of the association for computational linguistics: Human language technologies, proceedings of the conference, 19–24 June, 2011, Portland, Oregon, USA* (pp. 142–150).

Medhat, W., Yousef, A. H., & Mohamed, H. K. (2014). Component analysis of a sentiment analysis framework on different corpora. In *2014 9th international conference on computer engineering systems (icces)* (pp. 300–306). doi:10.1109/ICCES.2014.7030976.

Mejova, Y., & Srinivasan, P. (2011). Exploring feature definition and selection for sentiment classifiers. In L. A. Adamic, R. A. Baeza-Yates, & S. Counts (Eds.), *Proceedings of the fifth international conference on weblogs and social media, Barcelona, Catalonia, Spain, July 17–21, 2011* (pp. 546–549). The AAAI Press.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM, 38*(11), 39–41. doi:10.1145/219717.219748.

Mohammad, S., Kiritchenko, S., & Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the 7th international workshop on semantic evaluation, semeval@naacl-hlt 2013, Atlanta, Georgia, USA, June 14–15, 2013* (pp. 321–327).

Mohammad, S. M., Zhu, X., Kiritchenko, S., & Martin, J. D. (2015). Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing and Management, 51*(4), 480–499. doi:10.1016/j.ipm.2014.09.003.

Mullen, T., & Malouf, R. (2006). A preliminary investigation into sentiment analysis of informal political discourse. In *Computational approaches to analyzing weblogs, papers from the 2006 AAAI spring symposium, technical report ss-06-03, Stanford, California, USA, March 27–29, 2006* (pp. 159–162).

Na, J.-C., Sui, H., Khoo, C., Chan, S., & Zhou, Y. (2004). Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Conference of the international society for knowledge organization (isko)* (pp. 49–54).

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., & Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter. In *Second joint conference on lexical and computational semantics (*sem), volume 2: Proceedings of the seventh international workshop on semantic evaluation (semeval 2013)* (pp. 312–320). Atlanta, Georgia, USA: Association for Computational Linguistics.

Narayanan, V., Arora, I., & Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced Naive Bayes model. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, & X. Yao (Eds.), *Intelligent data engineering and automated learning - IDEAL 2013 - 14th international conference, IDEAL 2013, hefei, China, October 20–23, 2013. proceedings*. In *Lecture Notes in Computer Science: 8206* (pp. 194–201). Springer. doi:10.1007/978-3-642-41278-3_24.

Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications, 73*, 125–144. doi:10.1016/j.eswa.2016.12.036.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the international conference on language resources and evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta* (pp. 1320–1326).

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

Perkins, J. (2010). *Python text processing with NLTK 2.0 cookbook*. Packt Publishing.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137. doi:10.1108/eb046814.

Prasad, S. (2010). Micro-blogging Sentiment Analysis Using Bayesian Classification Methods. *Technical Report*. Stanford University.

Saif, H., Fernández, M., He, Y., & Alani, H. (2013). Evaluation datasets for twitter sentiment analysis: A survey and a new dataset, the sts-gold. In *Proceedings of the first international workshop on emotion and sentiment in social and expressive media: Approaches and perspectives from AI (ESSEM 2013) A workshop of the XIII international conference of the italian association for artificial intelligence (ai*ia 2013), Turin, Italy, December 3, 2013.* (pp. 9–21).

Saloot, M. A., Idris, N., Shuib, L., Raj, R. G., & Aw, A. (2015). Toward tweets normalization using maximum entropy. In *Proceedings of the workshop on noisy user-generated text, nut@ijcnlp 2015, Beijing, China, July 31, 2015* (pp. 19–27). doi:10.18653/v1/W15-4303.

Sánchez-Mirabal, P. A., Torres, Y. R., Alvarado, S. H., Gutiérrez, Y., Montoyo, A., & Muñoz, R. (2014). Umcc_dlsi: Sentiment analysis in twitter using polirity lexicons and tweet similarity. In P. Nakov, & T. Zesch (Eds.), *Proceedings of the 8th international workshop on semantic evaluation, semeval@coling 2014, Dublin, Ireland, August 23–24, 2014.* (pp. 727–731). The Association for Computer Linguistics.

dos Santos, C. N., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *COLING 2014, 25th international conference on computational linguistics, proceedings of the conference: Technical papers, August 23–29, 2014, Dublin, Ireland* (pp. 69–78).

Severyn, A., & Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, Santiago, Chile, August 9–13, 2015* (pp. 959–962). doi:10.1145/2766462.2767830.

Shotaroo, Takamura, H., & Okumura, M. (2005). Sentiment classification using word sub-sequences and dependency sub-trees. In *Advances in knowledge discovery and data mining, 9th pacific-asia conference, PAKDD 2005, Hanoi, Vietnam, May 18–20, 2005, proceedings* (pp. 301–311). doi:10.1007/1143091937.

Singh, T., & Kumari, M. (2016). Role of text pre-processing in twitter sentiment analysis. *Procedia Computer Science, 89*(Supplement C), 549–554. doi:10.1016/j.procs.2016.06.095. India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19–21, 2016, Bangalore, India

Symeonidis, S., Effrosynidis, D., Kordonis, J., & Arampatzis, A. (2017). DUTH at semeval-2017 task 4: A voting classification approach for twitter sentiment analysis. In *Proceedings of the 11th international workshop on semantic evaluation, semeval@acl 2017, Vancouver, Canada, August 3–4, 2017* (pp. 704–708). doi:10.18653/v1/S17-2117.

Tang, D., Wei, F., Qin, B., Liu, T., & Zhou, M. (2014). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation, semeval@coling 2014, Dublin, Ireland, August 23–24, 2014.* (pp. 208–212).

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, volume 1: Long papers* (pp. 1555–1565).

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *JASIST, 63*(1), 163–173. doi:10.1002/asi.21662.

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications, 57*, 117–126. doi:10.1016/j.eswa.2016.03.028.

Uysal, A. K., & Günal, S. (2014). The impact of preprocessing on text classification. *Information Processing and Management, 50*(1), 104–112. doi:10.1016/j.ipm.2013.08.006.

Wang, S. I., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *The 50th annual meeting of the association for computational linguistics, proceedings of the conference, July 8–14, 2012, Jeju island, Korea - volume 2: Short papers* (pp. 90–94).

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP 2005, human language technology conference and conference on empirical methods in natural language processing, proceedings of the conference, 6–8 october 2005, Vancouver, British Columbia, Canada* (pp. 347–354).

Wu, F., Huang, Y., & Yuan, Z. (2017). Domain-specific sentiment classification via fusing sentiment knowledge from multiple sources. *Information Fusion, 35*, 26–37. doi:10.1016/j.inffus.2016.09.001.

Wu, W., Zhang, B., & Ostendorf, M. (2010). Automatic generation of personalized annotation tags for twitter users. In *Human language technologies: Conference of the north american chapter of the association of computational linguistics, proceedings, June 2–4, 2010, Los Angeles, california, USA* (pp. 689–692). The Association for Computational Linguistics.

Xia, R., Xu, F., Zong, C., Li, Q., Qi, Y., & Li, T. (2015). Dual sentiment analysis: Considering two sides of one review. *IEEE Transactions on Knowledge and Data Engineering, 27*(8), 2120–2133. doi:10.1109/TKDE.2015.2407371.

Yamada, I., Takeda, H., & Takefuji, Y. (2015). Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the workshop on noisy user-generated text, nut@ijcnlp 2015, Beijing, China, July 31, 2015* (pp. 136–140). doi:10.18653/v1/W15-4320.

Zhang, Z., Wu, G., & Lan, M. (2015). ECNU: multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. In D. M. Cer, D. Jurgens, P. Nakov, & T. Zesch (Eds.), *Proceedings of the 9th international workshop on semantic evaluation, semeval@naacl-hlt 2015, Denver, Colorado, USA, June 4–5, 2015* (pp. 561–567). The Association for Computer Linguistics.

Zhao, J. (2015). Pre-processing boosting twitter sentiment analysis? In *2015 IEEE international conference on smart city/socialcom/sustaincom 2015, Chengdu, China, December 19–21, 2015* (pp. 748–753). IEEE Computer Society. doi:10.1109/SmartCity.2015.158.

Zhao, J., & Gui, X. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access, 5*, 2870–2879. doi:10.1109/ACCESS.2017.2672677.