A Comparison of Pre-processing Techniques for Twitter Sentiment Analysis

Dimitrios Effrosynidis[(✉)], Symeon Symeonidis, and Avi Arampatzis

Database and Information Retrieval Research Unit,
Department of Electrical and Computer Engineering,
Democritus University of Thrace, 67100 Xanthi, Greece
{dimievfr,ssymeoni,avi}@ee.duth.gr
http://www.nonrelevant.net

Abstract. Pre-processing is considered to be the first step in text classification, and choosing the right pre-processing techniques can improve classification effectiveness. We experimentally compare 15 commonly used pre-processing techniques on two Twitter datasets. We employ three different machine learning algorithms, namely, Linear SVC, Bernoulli Naïve Bayes, and Logistic Regression, and report the classification accuracy and the resulting number of features for each pre-processing technique. Finally, based on our results, we categorize these techniques based on their performance. We find that techniques like stemming, removing numbers, and replacing elongated words improve accuracy, while others like removing punctuation do not.

Keywords: Sentiment analysis · Text pre-processing · Machine learning · Text classification

1 Introduction

In the last decade, Sentiment Analysis in microblogging has become a very popular research area. People share their daily life with messages in platforms such as Twitter, and posts of users are related with many topics. Many studies present interesting approaches for classification methods in sentiment analysis, e.g. [1,9], and refer to the important role of pre-processing before and during the feature selection process.

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

According to [4], the total percentage of noise in a dataset reaches 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 use punctuation signs to emphasize their emotions,

J. Kamps et al. (Eds.): TPDL 2017, LNCS 10450, pp. 394–406, 2017.
DOI: 10.1007/978-3-319-67008-9_31

like 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, it arises the need of cleansing and normalizing the data, 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 many common pre-processing techniques from other previous studies, plus a few novel ones 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 on two well-known datasets. In the end, based on our results, we suggest to future researchers which techniques are more suitable for Twitter sentiment analysis and which have to be avoided.

The rest of this paper is structured as follows. The following section includes a review of the related literature. Section 3 presents the pre-processing techniques that we will compare. Section 4 describes the datasets, the machine learning algorithms, and the evaluation methodology. Results and conclusions are discussed in Sects. 5 and 6, respectively.

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 to demonstrate the difference between these techniques and their contribution to the final result of classification.

In [19], the authors 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. They remove URLs, hashtags, user mentions, punctuation, and stopwords, and they identify the importance of slang words and spelling correction. They use an SVM classifier in their experiments.

The role of pre-processing is also investigated by [18] 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. They also use an SVM classifier and correlate the number of features to its accuracy. They show 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 [21] for two languages on e-mails and news. They use stopword removal, lowercase conversion, and stemming, and they evaluate with micro-F_1 score using an SVM classifier. They show that there is no unique combination of pre-processing techniques that improves accuracy on any domain or language and that researchers should carefully analyze all possible combinations.

There is also a workshop named ‘Workshop on Noisy User-generated Text’[1], that is running since 2015 and focuses on natural language processing applied to noisy user-generated text that is found online. In 2015, they introduced a lexical normalization task, in aiming to normalise non-standard words in English Twitter messages to their canonical forms.

Thus, many studies have examined the role of pre-processing, generally and specifically in sentiment analysis, however, none of them has gathered in a comparative study the total number of techniques which will be presented in Sect. 4.

3 Common Pre-processing Techniques

Below we describe the 15 pre-processing techniques we will experiment with.

Remove Numbers. It is a common tactic to remove numbers from text, because they do not contain any sentiment. However, some researchers argue that keeping the numbers may improve classification effectiveness [6].

Replace 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 replace with ‘multiQuestionMark’.

Handling Capitalized Words. Same as before, capitalized words may imply intense emotion, so we detect all the words that are longer than two characters with all of their characters capitalized. We prefix them with ‘ALL_CAPS_’ like [16] did, so they can be identified in machine learning.

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

Replace Slang and Abbreviations. Social media users usually write in an informal way and their texts contain a lot of slang and abbreviations. These words, 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 their replacements. Some examples are the words ‘ty’, ‘qq’ and ‘omg’, which respectively mean and replaced by ‘thank you’, ‘crying’, and ‘oh my god’.

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

Replace Elongated Words. Elongated is a word when it contains a character that is repeating more than two times, like the word 'greeeeat'. It is important 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 because of their low frequency of occurrence. Detecting and replacing elongated words have been examined by researchers before, e.g. in [8].

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

Replace Negations with Antonyms. It is an approach that has not been used by many researchers and is presented in [14]. We search in each sentence for the word 'not' and then, we check if the next word has an antonym. If yes, we replace both words with the antonym. For example, the phrase 'not good' will be replaced with the word 'bad', using WordNet [7].

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

Remove Stopwords. Stopwords are function words with high frequency of presence across all sentences. It is considered needless to analyze them, because they do not contain much useful information. 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. In our implementation, we used the standard stopwords provided by NLTK [2].

Stemming. It is the process of removing the endings of the words in order to detect their root form. By doing so, many words are merged and the dimensionality is reduced. It is a widely used method that generally provides good results; we used the Porter Stemmer [15].

Lemmatizing. Another method of merging many words to one is Lemmatization. In this method, we remove the endings of the words in order to detect their lemmas, i.e. their root forms in a dictionary.

Replace URLs and User Mentions. In Twitter texts, almost every sentence contains a URL and a user mention. Their presence does not contain any sentiment and one approach is to replace them in pre-processing with tags as [1] did. We used the tags 'URL' and 'AT_USER'.

Spelling Correction. It is very common in informal texts for users to make spelling errors that might make classification harder. By using tools that automatically correct these errors, it is possible to improve classification effectiveness [10]. While no corrector is perfect, they have some —usually high— accuracy of success. We used Norvig's spelling corrector.[2]

Remove Punctuation. In many works, it is common to remove punctuation signs in pre-processing [6]. However, many times the presence of punctuation marks denotes the existence of some sentiment. For example, an exclamation mark may mean an intense positive or negative sentiment. So if we remove them we might decrease the accuracy of classification.

Table 1. Correspondence of pre-processing techniques

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

Table 1 summarizes 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 labels 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 in [17]. We chose to examine the three-point classification problem with the predefined classes of positive, negative, and neutral. For this task, we used two datasets, the first being the Sentiment Strength Twitter dataset and the second the SemEval dataset, both described next.

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

4.1 The Sentiment Strength Twitter Dataset

The Sentiment Strength Twitter or SS-Twitter dataset contains 4,242 tweets and was developed by [20] in order to evaluate SentiStrength[3], a lexicon-based method for sentiment strength detection. The tweets are labeled 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, we created a new one with three sentiment labels (positive, negative, neutral), suitable for our task. Hence, we apply two rules, as done in [17]. 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, and negative otherwise. After these transformations, the final dataset consists of 1,252 positive, 1,037 negative and 1,953 neutral tweets. Some statistics related to the dataset are shown in Table 2.

Table 2. Statistics of the datasets

	SS-Twitter	SemEval
Total sentences	4,242	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	3,467	34,979
Total slangs	622	5,815
Total elongated words	1,543	17,355
Total multi exclamation marks	325	2,834
Total multi question marks	152	750
Total multi stop marks	1,118	14,115
Total all capitalized words	2,854	52,141

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 sentiment analysis in three-point classification. Each tweet was manually annotated by Amazon Mechanical Turk workers or CrowdFlower users, depending on the year. This task is running each year since 2013 [12], and every year more data are added. By collecting the datasets of all years (2013–2017), we gathered 65,854 tweets, i.e. 23,197 positive, 12,510 negative, and 30,147 neutral. Some statistics related to this dataset are also shown in Table 2.

[3] http://sentistrength.wlv.ac.uk.

[4] http://alt.qcri.org/semeval2017/.

4.3 Machine Learning Algorithms

Out of the many available supervised machine learning algorithms, we chose one algorithm for each of the three most used categories. These are, the Generalized Linear Models (GLM), the Naïve Bayes (NB), and the Support Vector Machines (SVM). From the GLM family we chose the Logistic Regression algorithm, from the NB we chose the Bernoulli Naïve Bayes, and from the SVMs we chose the Linear SVC algorithm.

Logistic Regression. 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 [13].

Bernoulli Naïve Bayes. Naïve Bayes algorithms are the simplest probabilistic classification algorithms [5] that are widely used in sentiment analysis. They are based on the Bayes Theorem, which assumes a complete independence of variables. The Bernoulli algorithm is an alternative of Naïve Bayes, where 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.

Linear SVC. One of the most popular machine learning methods for classification of linear problems are SVMs [3]. They try to find a set of hyperplanes that separate the space into dimensions representing classes. These hyperplanes are chosen in a way to maximize 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.

All the models that have been selected are in fact linear. 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.

4.4 Feature Extraction and Evaluation

There are several ways to assess the features in a bag-of-words representation. We chose to use Term Frequency – Inverse Document Frequency (TF.IDF) which is given by

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

where f is the number of occurrences in the document, N is the number of documents, and df is the number of documents that contain this feature [11].

The metric that was used to evaluate the classification results is accuracy, which is the number of the correct classifications out of all classifications. Accuracy is a good metric for balanced datasets like in our case. Finally, we used uni-grams, and compare the numbers of resulting features across pre-processing methods.

5 Results

In this section, we present the results of the use of every pre-processing technique among the two datasets and between the three classifiers.

With a dataset as input, we used Python's NLTK [2] and created a new file as output for each pre-processing technique. 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.

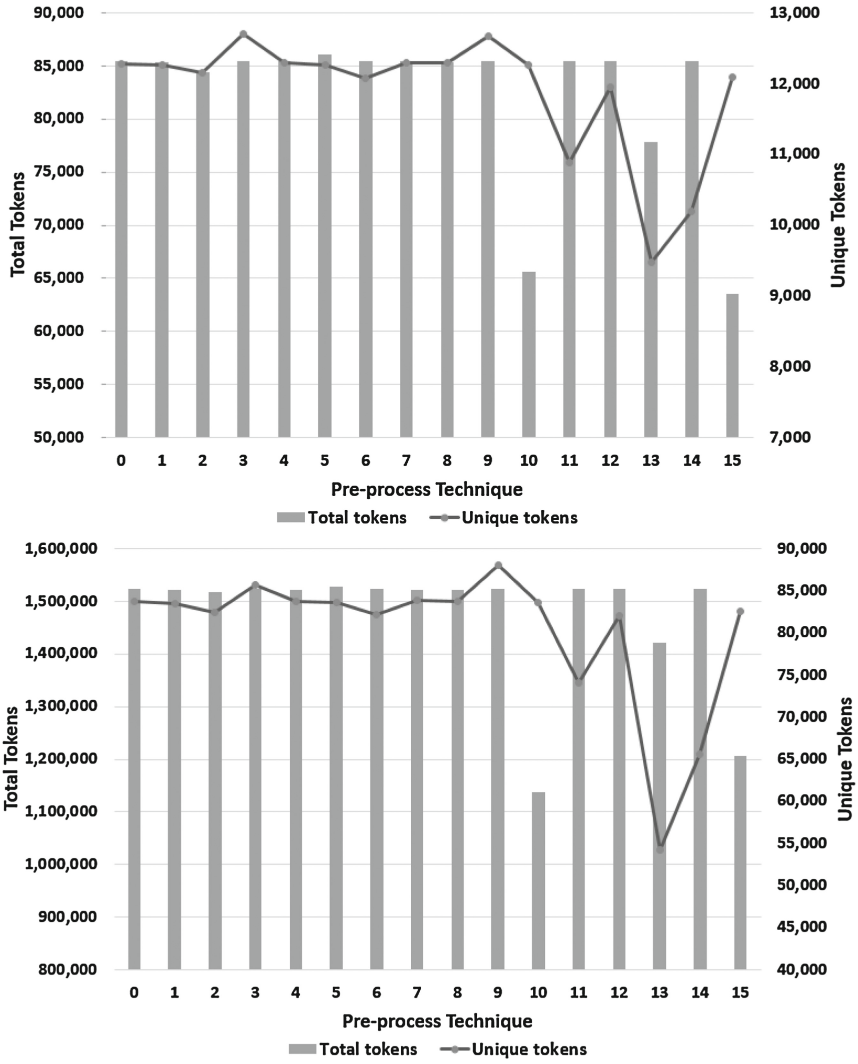

Fig. 1. Total and unique tokens per pre-processing technique in SS-Twitter (above) and SemEval (below) datasets

As we can see, handling negations by adding the 'NOT_' prefix in front of the words (technique 9), results in an augmentation of both the total and the unique tokens, with the latter being clearly visible. The other technique that increases the total tokens is the slang and abbreviation replacement (technique 5), but it decreases the unique tokens. Removing stopwords (technique 10) and punctuation (technique 15) results in a great reduction for the total tokens, but not a remarkable reduction on the unique tokens. The three techniques that reduce a lot the unique tokens are stemming, the replacement of URLs and user mentions, and spelling correction (techniques 11, 13, and 14). Both datasets present the same proportions in the total and unique tokens.

The number of unique tokens defines the number of features that will be used in a uni-gram bag-of-words representation. The quality and number of these features play a key role in the accuracy of the classifiers. Keeping a significant number of words/features will increase the temporal and spatial complexity of classifiers. This also favors the appearance of overfitting. Although, an increase in the number of features may not always result in better classification, because the quality of features also matters.

After the creation of the new pre-processed files, we apply machine learning algorithms using Sklearn [13]. For vectorization we used the tf-idf transformation and as features we utilized uni-grams, so we can see if and how the number of features has an impact on the classification results. As said before, we chose three representative algorithms (Linear SVC, Bernoulli Naïve Bayes, and Logistic Regression) and we did not make any changes to their parameters. The results for both datasets are presented in Fig. 2. For each pre-processing technique we compare the accuracy and the number of features per three classifiers.

For the SS-Twitter dataset we observe that the techniques which result in increased accuracy in all classifiers are 1, 2, 6, 7, 11 and 12. The highest results were 61.4% for the Linear SVC which was achieved by replacing the elongated words, 60.6% for the Bernoulli Naïve Bayes which was achieved by stemming, and 61% for the Logistic Regression which was achieved by using lowercase. The lowest accuracy for all classifiers occurs when we remove punctuation signs (technique 15), showing their importance in sentiment classification. Other poorly performing techniques were 3, 5, 8, 10, and 14, which only resulted in a small increase in one classifier. Finally, the techniques 4, 9, and 13, resulted in an increment in two classifiers and can be considered good techniques.

For the SemEval dataset, the techniques which provide better accuracy than the initial for all classifiers are 1, 2, 11 and 13. Especially the latter, which is the replacement of URLs and user mentions, gives the highest results with 59% for Linear SVC, 60.6% for Bernoulli Naïve Bayes, and 60.7% for Logistic Regression. The lowest accuracy is noticed when we apply the techniques 5, 10, 14, and 15 for all classifiers. The poorly performing techniques in this dataset are 3, 4, 6, 7 and 8. Finally, other highly performing techniques which result in improved accuracy in two classifiers are 8, 9, and 12.

Based on the results, we can discern 5 categories depending on the accuracy. These categories describe how the SS-Twitter and the SemEval datasets reacted

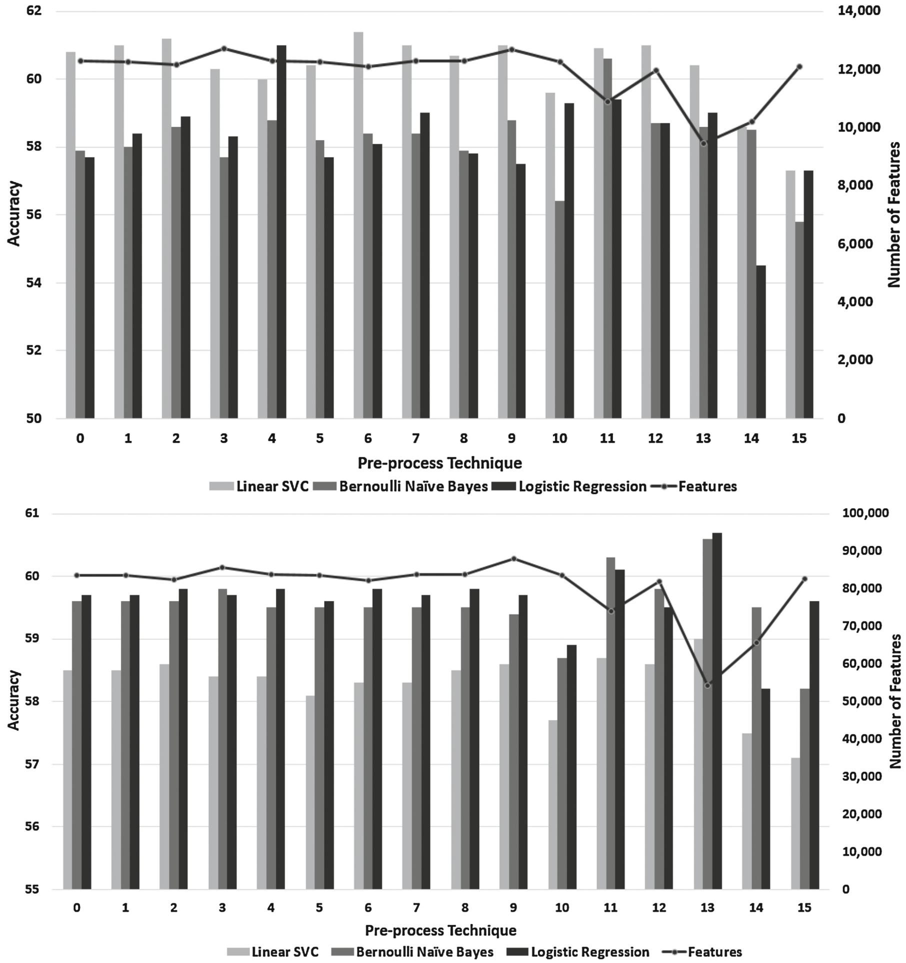

Fig. 2. Accuracy percentage and number of features for all pre-processing techniques per three machine learning algorithms in SS-Twitter (above) SemEval (below) datasets

to the 15 pre-processing techniques for three-point Twitter sentiment analysis and are presented in Table 3.

We note that there is no significant association between the number of features and the accuracy. The techniques that increase the number of features are 3, 4, 7, 8, and 9 and only one of them has high performance. The techniques which achieve a great reduction of features like 11, 13, and 14, give better results. The rest reduce the features by a few, and their accuracy varies.

Table 3. Accuracy performance categories for all pre-processing techniques on both datasets

Performance	Description	Techniques
Best	High accuracy in all classifiers and all datasets	1, 2, 11
High	High accuracy in most classifiers and all datasets	9, 12, 13
Poor	Low accuracy in most classifiers and all datasets	3, 5, 8, 10, 14
Worst	Lowest accuracy in all classifiers and all datasets	15
Varying	High or poor accuracy in most classifiers depending on the dataset	4, 6, 7

6 Conclusion and Future Work

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 were not evaluated in a comparative study in the past, and tested them in two datasets. Each technique was evaluated in three representative machine learning algorithms on accuracy. Finally, we distinguish some performance categories based on the results and count the number of features for each technique.

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 stemming, replacement of repetitions of punctuation, and removing numbers. The non-recommended techniques include removing punctuation, 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. Thus, in future work, we will extend our analysis with more machine learning algorithms and we will try to combine these techniques to achieve better results. Moreover, another future approach is to test these techniques on datasets from different domains such as news articles and product or movie reviews.

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media, LSM 2011, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 30–38 (2011). http://dl.acm.org/citation.cfm?id=2021109.2021114
2. Bird, S.: NLTK: the natural language toolkit. In: Calzolari, N., Cardie, C., Isabelle, P. (eds.) ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17–21 July 2006. The Association for Computer Linguistics (2006). http://aclweb.org/anthology/p06-4018

3. Cherkassky, V.: The nature of statistical learning theory. IEEE Trans. Neural Netw. **8**(6), 1564 (1997). doi:10.1109/TNN.1997.641482
4. Fayyad, U.M., Piatetsky-Shapiro, G., Uthurusamy, R.: Summary from the KDD-03 panel: data mining: the next 10 years. SIGKDD Explor. **5**(2), 191–196 (2003). doi:10.1145/980972.981004
5. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: UAI 1995: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Canada, 18–20 August 1995, pp. 338–345 (1995). https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=450&proceeding_id=11
6. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, 2–6 November 2009, pp. 375–384 (2009). http://doi.acm.org/10.1145/1645953.1646003
7. Miller, G.A.: WordNet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995). doi:10.1145/219717.219748
8. Mohammad, S., Kiritchenko, S., Zhu, X.: 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, 14–15 June 2013, pp. 321–327 (2013). http://aclweb.org/anthology/S/S13/S13-2053.pdf
9. Mohammad, S.M., Zhu, X., Kiritchenko, S., Martin, J.D.: Sentiment, emotion, purpose, and style in electoral tweets. Inf. Process. Manage. **51**(4), 480–499 (2015). doi:10.1016/j.ipm.2014.09.003
10. Mullen, T., Malouf, R.: 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, 27–29 March 2006, pp. 159–162 (2006). http://www.aaai.org/Library/Symposia/Spring/2006/ss06-03-031.php
11. Na, J.C., Sui, H., Khoo, C., Chan, S., Zhou, Y.: 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 (2004)
12. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., Wilson, T.: SemEval-2013 task 2: sentiment analysis in twitter. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, pp. 312–320. Association for Computational Linguistics, Atlanta, Georgia, USA, June 2013. http://www.aclweb.org/anthology/S13-2052
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011). http://dl.acm.org/citation.cfm?id=2078195
14. Perkins, J.: Python Text Processing with NLTK 2.0 Cookbook. Packt Publishing, Birmingham (2010)
15. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980). doi:10.1108/eb046814
16. Prasad, S.: Micro-blogging sentiment analysis using bayesian classification methods. Technical report (2010)

17. Saif, H., Fernández, M., He, Y., Alani, H.: 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, 3 December 2013, pp. 9–21 (2013). http://ceur-ws.org/Vol-1096/paper1.pdf
18. Shi, Y., Xi, Y., Wolcott, P., Tian, Y., Li, J., Berg, D., Chen, Z., Herrera-Viedma, E., Kou, G., Lee, H., Peng, Y., Yu, L. (eds.): Proceedings of the First International Conference on Information Technology and Quantitative Management, ITQM 2013, Dushu Lake Hotel, Sushou, China, 16–18 May 2013, Procedia Computer Science, vol. 17. Elsevier (2013). http://www.sciencedirect.com/science/journal/18770509/17
19. Singh, T., Kumari, M.: Role of text pre-processing in twitter sentiment analysis. Proc. Comput. Sci. **89**, 549–554 (2016). http://www.sciencedirect.com/science/article/pii/S1877050916311607
20. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. JASIST **63**(1), 163–173 (2012). doi:10.1002/asi.21662
21. Uysal, A.K., Günal, S.: The impact of preprocessing on text classification. Inf. Process. Manage. **50**(1), 104–112 (2014). doi:10.1016/j.ipm.2013.08.006