On the Reusability of Sentiment Analysis Datasets in Applications with Dissimilar Contexts

S. Sarlis and I. Maglogiannis(✉)

Department of Digital Systems, University of Piraeus, Piraeus, Greece
stevensarlis@gmail.com, imaglo@unipi.gr

Abstract. The main goal of this paper is to evaluate the usability of several algorithms on various sentiment-labeled datasets. The process of creating good semantic vector representations for textual data is considered a very demanding task for the research community. The first and most important step of a Natural Language Processing (NLP) system, is text preprocessing, which greatly affects the overall accuracy of the classification algorithms. In this work, two vector space models are created, and a study consisting of a variety of algorithms, is performed on them. The work is based on the IMDb dataset which contains movie reviews along with their associated labels (positive or negative). The goal is to obtain the model with the highest accuracy and the best generalization. To measure how well these models generalize in other domains, several datasets, which are further analyzed later, are used.

1 Introduction

Sentiment analysis, also known as opinion mining, refers to the subjective analysis of a text [1]. Its basic task is predicting the overall sentiment polarity of a given sentence. In more recent works [2], sentiment analysis expands this task on trying to identify the emotional status of a sentence (e.g., sadness, excitement, joy, anger). It mainly applies to reviews, social media posts and survey responses, helping businesses understand their customer's opinions about their products and services, improving their marketing strategies and decision policies [3]. This practice of extracting insights from data is widely adopted by organizations and enterprises, affecting many aspects of human life.

Every day, many positive and negative comments are shared on the Internet. The impact of these comments on the economy is palpable and has led to the implementation of techniques that take advantage and extract knowledge from them. The sentiment of a movie review is usually associated with a distinct rating, which can be used for classification problems. From a user's perspective, it can be used as a recommendation tool for movie selection [4].

The IMDb dataset contains 50,000 reviews and is evenly divided into 25,000 train reviews and 25,000 test reviews. Movie reviews are labeled as positive or negative, and the challenge is to predict the sentiment of an unseen review. In this work, the

I. Maglogiannis et al. (Eds.): AIAI 2020, IFIP AICT 583, pp. 409–418, 2020.
https://doi.org/10.1007/978-3-030-49161-1_34

following classifiers are used: Logistic Regression, Bernoulli Naïve Bayes, Multinomial Naïve Bayes, Linear Support Vector Machine and two Neural Networks. Different techniques, including Count Vectorizer, TF-IDF (Term Frequency – Inverse Document Frequency) Vectorizer, lemmatization, stopwords, n-grams, minimum-maximum number of words and max features are implemented. The experiments conducted show that Logistic Regression, Linear Support Vector Machine and Neural Network classifiers perform noticeably better than Bernoulli Naïve Bayes and Multinomial Naïve Bayes.

The study aims to find answers to the following three questions:

1. How various classification algorithms perform on the IMDb sentiment analysis classification problem?
2. How much pre-processing improves the performance of these classifiers?
3. Do these classifiers generalize well to other datasets and can they therefore be used in different real-life problems?

The remainder of this paper is structured in four sections, as follows: Sect. 2 presents the background information and related research work, while Sect. 3 proposes the sentiment analysis system architecture. Section 4 describes the system in practice and reports the experiments conducted and the corresponding results. Finally, Sect. 5 concludes the paper and provides future work ideas.

2 Background and Related Work

Several approaches have been proposed for the extraction of features from text and the exploitation of them in forming appropriate classifying models. Initial attempts in order to represent a document used Bag of Words (BOW) representations, where the document was represented as the sum of one-hot vectors of its words. However, in the field of NLP, state-of-the-art Word Embeddings outperform BOW [5], as BOW representations lose some of the semantic meaning associated with text [6]. Therefore, more recent works create vector representations for both words and documents, to retain as much information as possible. Furthermore, word embeddings can boost the generalization capabilities of neural systems [7, 8]. However, in this work, BOW representations are used, as they require just a few lines of code to build the models and they work just fine in these specific datasets. The scikit-learn library, incorporated in Python [9], provides different schemes that can be used, including word counts with Count Vectorizer and word frequencies with TF-IDF (Term Frequency – Inverse Document Frequency) Vectorizer, which are further analyzed in this work. Moreover, [10] presented the Delta TF-IDF technique, which is considered to be an improvement of the TF-IDF. The difference is that in regular TF-IDF, each word or combination of words, is associated with a TF value (word counts in the document) and an IDF value (word counts in all documents), while delta TF-IDF gives more weight to words that occur more often in that text, and are rarer in oppositely labeled documents. It is shown that the Delta TF-IDF produces noticeably better results than the regular TF-IDF. As mentioned earlier, text preprocessing is considered the most important step for NLP tasks. In an NLP pipeline, many preprocessing techniques are used including but not limited to lowercasing, lemmatization, character removal. These techniques have already been studied in [3] and they are further analyzed later in this work.

3 Proposed Methodology for Sentiment Analysis

The process steps and the workflow diagram used for the sentiment analysis is shown in Fig. 1. Firstly, the IMDb dataset is collected and imported. Subsequently, in order to remove noise and clean the data, pre-processing is performed. The next step is feature extraction and feature selection, where each document is represented as a document-term matrix. In this matrix, rows refer to documents in the collection and columns correspond to terms. In the final step, the classification process is performed, and the used algorithms are evaluated.

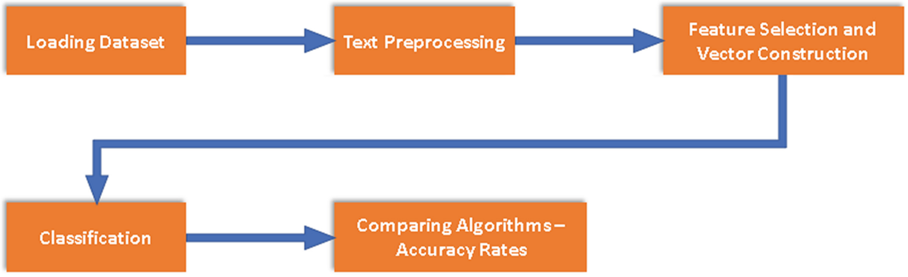
Loading Dataset
Text Preprocessing
Feature Selection and Vector Construction
Classification
Comparing Algorithms – Accuracy Rates

Fig. 1. Workflow diagram

3.1 Text Preprocessing

Before performing any classification algorithm, it is important to clean up the data. Finding and removing noise, greatly affects the accuracy of the classification algorithms. The following preprocessing techniques are applied: lowercasing, lemmatizing and removing noise characters and stop words. Lowercasing is probably the simplest and most effective preprocessing technique, where words map to the same lowercase form. This technique applies to all characters of the input text. Furthermore, the reviews are cleaned by removing HTML tags, non-word characters, punctuation and accents, which do not serve any purpose for detecting sentiment. Subsequently, a set of commonly used, low information words (i.e., stopwords), which do not carry any sentiment, is removed. During the stopwords process, the words included in the "NLTK list of English stopwords" are removed, except for the words "no" and "not" which are excluded, as they affect the value of the label. This initiative stems from the idea that, for example, the expression "not good" expresses a negative feeling, whereas if the word "not" was removed, the expression would become "good", which has a positive connotation, thus distorting the result. The process of lemmatization replaces the various forms a word can have, with its corresponding lemma (i.e., cutting out endings). This process is similar to stemming but it is preferred, as it doesn't just chop things off, but it actually transforms words to their actual root. Lemmatization is a standard preprocessing technique for linear text classification systems and has been shown to improve classification accuracy in sentiment analysis tasks [11]. Finally, n-grams and more specifically bi-grams are used in some models. N-grams are a contiguous sequence of n items and consequently bi-grams are a sequence of 2 words [12].

3.2 Feature Selection and Feature Extraction (Vectorization)

Common machine learning algorithms expect a two-dimensional table as an input, where rows are instances and columns are attributes. In this context, documents must be converted into vector representations. This process is called "feature extraction" or "vectorization" and is an important step when analyzing text. Words are coded as integers or as floating-point values, depending on which technique is implemented and are then used as an input to machine learning algorithms. In this work, two of the most well-known such techniques are used: The Count Vectorizer technique and the TF-IDF Vectorizer technique [13].

Moreover, in order to apply machine learning algorithms on text segments, "feature selection" may be need, so as to reduce the input dimension, since the datasets may be very large and contain many words. Even after the preprocessing and the cleaning of the data, there may be a lot of different words in the dataset, which create a dictionary. Depending on the selected technique, Count Vectorizer or TF-IDF Vectorizer, these words are sorted based on their importance.

3.2.1 Count Vectorizer

Count Vectorizer is one of the simplest and relatively satisfactory techniques. It counts the number of times a word appears in a document and uses this value as its weight. The Count Vectorizer technique provides tokenization of the text documents and builds a vocabulary of words. Count Vectorizer counts words, so it returns integers, while TF-IDF Vectorizer assigns a score to words, so it returns floats.

3.2.2 Term Frequency and Inverse Document Frequency (TF-IDF) Vectorizer

In the TF-IDF Vectorizer technique, the weight corresponding to each word not only depends on its frequency in the document, but also on how repetitive this word is in all documents. TF-IDF is a statistic that aims to reflect a word's importance in a document within a collection of documents [14]. This technique is mainly used in information retrieval in order to rank how important a keyword is to a given document in a corpus and is a common approach for sentiment analysis [15]. Term Frequency (TF) is the frequency of the term within a document and indicates how many times a word is used within this document [14]. TF describes how important a word is in the document, but it doesn't consider the possibility of that word occurring to other documents too, thus a balance term is needed. The Inverse Document Frequency (IDF) refers to the specificity of a term, which can be quantified as an inverse function of the number of documents in which it occurs. The more documents that have a specific word, the lower its IDF score is.

3.3 Classification and Comparison

To solve the classification problem, the following algorithms are used: Logistic Regression (LR), Bernoulli Naïve Bayes (BNB), Multinomial Naïve Bayes (MNB), Linear Support Vector Machine (Linear SVM) and two Neural Networks (NN1 & NN2). All the parameters of the classifiers use the default values provided by the scikit-learn package, except for the Logistic Regression and the Linear SVM, where in the Count Vectorizer technique, the value of the max interactions parameter is set to 1,000 and 10,000 respectively. Furthermore, the Linear SVM classifier uses the hinge loss function, which is the standard SVM loss. Regarding the architecture of the Neural Networks, they are Sequential Neural Networks with four Dense layers. The first three Dense layers use the "relu" activation function and the last uses the "sigmoid". The Networks use two Dropout layers to avoid overfitting, "killing" random nodes each time, so that the network can recalibrate its weights. The optimizer is "adam" and the loss function is binary cross entropy. The batch size is set to 500 and the epochs used are only two, as the model overfits if trained longer. The two networks need to know what input shape they should expect. In this work, they have different input shapes. More specifically, the NN1 has an input shape equal to 57,210 and the NN2 has an input shape equal to 50,000. This results from the fact that in the NN1 the two vectorization techniques (Count and TF-IDF Vectorizer) are implemented with the first approach, which does not consider max features, while in the NN2 the two techniques are implemented with the second approach, which considers max features (50,000) and bi-grams. Max features refer to the number of words or combinations of words required, to construct the feature vectors. The metric used to evaluate these classifiers is accuracy, as there is an equal number of samples belonging to each class [16]. Moreover, the models are trained on the IMDb dataset and tested on a variety of domains including food, clothing, tweet and hotel datasets. The IMDb is selected as the training dataset, as it contains more than 83 million registered users [4], providing a rich and diverse source of human sentiments.

4 Experimentation and Results

This section reports the experiments conducted and the corresponding results. More specifically, in this section, the evaluation of the performance of these algorithms and their ability to generalize to other sentiment recognition problems is explored.

4.1 Description of Problems and Datasets

The IMDb dataset contains 50,000 reviews and is divided into 25,000 train and 25,000 test reviews. Movie reviews are labeled as positive or negative. The first problem is a classification problem, in which a variety of classification algorithms is being evaluated. After preprocessing and cleaning the data, there are 57,210 different words in the train set (25,000 reviews). Since there is no limitation on the run-time of the algorithms and the memory size, a large set of words was chosen, which leads to better accuracy rates. Therefore, for every classifier, the Count Vectorizer and the TF-IDF Vectorizer techniques are implemented. The main contribution of this work is to examine the generalization of the trained with the IMDb dataset classifiers on other sentiment recognition problems. A large set of words is chosen, in order to see if the classification models can generalize well to other datasets, since with larger sets of words (i.e., dictionaries) it is more likely to achieve better generalization results. More specifically, two different approaches are implemented. The first, takes all the 57,210 words for constructing the feature vectors. The second one, takes the top 50,000 words or combinations of words (i.e., bi-grams) for constructing the feature vectors. The datasets used to measure how well these models generalize include Women's E-Commerce Clothing Reviews Dataset [17], Amazon Fine Food Reviews Dataset [18], Datafiniti's Business Hotel Reviews Dataset [19], Consumer Reviews of Amazon Products Dataset [20] and two Twitter Datasets [21, 22]. The first four datasets have a score column with values between 1 to 5, therefore in order to determine if a sentiment is positive or negative, reviews with score $\geq$4 are mapped to label 1 (positive review) and reviews with score $\leq$2 are mapped to label 0 (negative review). The Twitter datasets are ready to be used directly in binary classification models (0 for negative and 1 for positive sentiment) (Table 1).

Table 1. Datasets utilized in this work and corresponding samples.

Datasets	Test samples
IMDb large movie review dataset v1.0	25,000
Women's e-commerce clothing reviews	20,615
Amazon fine food reviews	36,813
Datafiniti's business hotel reviews	8,438
Consumer reviews of Amazon products	33,128
Tweets dataset - Ayoub Benaissa	99,989
Sentiment140 dataset with 1.6 million tweets	100,000

4.2 Results

As already mentioned, the IMDb dataset contains 50,000 reviews and the 25,000 of them are used to train the classification models. The other four datasets consisted of more test samples but since their label was mapped to 1 and 0 if they had score $\geq$4 or score $\leq$2 respectively, records with score equal to 3 were ignored, as they did not serve any purpose. The first Twitter dataset was fully used and from the second Twitter dataset 1/16 of the data was randomly selected. The results are shown in Tables 2 and 3 and they are derived from the following classification algorithms: Logistic Regression (LR), Bernoulli Naïve Bayes (BNB), Multinomial Naïve Bayes (MNB), Linear Support Vector Machine (Linear SVM), Neural Network (NN1) and the following vectorization techniques: Count Vectorizer, TF-IDF Vectorizer. These two vectorization techniques are also implemented with the second approach, which considers max features (50,000) and bi-grams and is tested on the second Neural Network (NN2). The results on the IMDb Dataset concern the 25,000 reviews used for testing.

Table 2. The accuracy of the models using the **Count Vectorizer technique**.

Classifier	Experiments						
	Trained on this set (50% split)	IMDb trained classifiers evaluated on various datasets					
	IMDb dataset	Clothing dataset	Food dataset	Hotel dataset	Product dataset	1st Twitter dataset	2nd Twitter dataset
Logistic Regression	0.85824	0.80392	**0.75470**	**0.77281**	**0.84876**	**0.59829**	0.58368
Bernoulli Naïve Bayes	0.8196	0.45651	0.40895	0.57703	0.29890	0.45524	0.51936
Multinomial Naïve Bayes	0.82076	0.70972	0.5827	**0.77447**	0.75066	0.57522	0.58787
Linear SVM	0.8318	0.69876	0.70211	0.71284	0.80919	0.57536	0.56601
Neural Network I	**0.8769**	**0.8385**	0.7287	**0.7817**	**0.8405**	**0.6043**	**0.5979**
Neural Network II	**0.8865**	**0.8432**	**0.7659**	**0.7687**	0.8156	0.5774	**0.5907**

Table 3. The accuracy of the models using the **TF-IDF Vectorizer technique**.

Classifier	Experiments						
	Trained on this set (50% split)	IMDb trained classifiers evaluated on various datasets					
	IMDb dataset	Clothing dataset	Food dataset	Hotel dataset	Product dataset	1st Twitter dataset	2nd Twitter dataset
Logistic Regression	**0.87996**	**0.84826**	**0.78792**	**0.82223**	**0.84964**	**0.61008**	**0.6053**
Bernoulli Naïve Bayes	0.8196	0.45651	0.40895	0.57703	0.29890	0.45524	0.51936
Multinomial Naïve Bayes	0.82744	0.69721	0.57618	0.73631	0.73025	0.56942	0.58517
Linear SVM	**0.87516**	**0.83371**	0.74568	0.78916	0.81010	**0.59899**	**0.59454**
Neural Network I	**0.8786**	0.7392	0.6747	0.7234	0.7479	0.5800	**0.5952**
Neural Network II	**0.8888**	**0.8488**	**0.7776**	**0.8072**	**0.8310**	**0.5997**	**0.5944**

4.3 Discussion on the Results

From the experimental results listed above, the following conclusions arise:

1. In each model, TF-IDF technique provides higher accuracy rates, except for Multinomial Naïve Bayes and Neural Network I, where Count Vectorizer technique gives higher percentages. More specifically, in these two models, TF-IDF is better on the IMDb test set, resulting from the dataset in which the models have been trained and worse in other datasets. This means that it does not generalize as good as the other technique.

    a. The Multinomial classifier is used when features have discrete values. In practice, fractional counts such as TF-IDF also work [23], however they lose some accuracy, as shown in the results.
    b. As far as the Neural Network I is concerned, which approaches the task by taking all the 57,210 words to construct the feature vectors, Count Vectorizer outperforms TF-IDF Vectorizer. This approach is followed by all the models except for the Neural Network II, which takes the max features (50,000) and bi-grams.

2. The Bernoulli Naïve Bayes classifier gives the same accuracy rates for both techniques, in each dataset. Bernoulli Naïve Bayes is used for features with binary or boolean values. Therefore, TF-IDF values are converted to 0 or 1 (e.g., float values equal to zero are mapped to zero and float values greater than zero are mapped to one), hence, to Count Vectorizer values. The comparison of Bernoulli and Multinomial models was conducted, since training Naïve Bayes models is fairly cheap.
3. Lastly, neither of the classifiers performs good on the two Twitter datasets. This is due to the fact that when analyzing tweets, different preprocessing techniques should be used. In particular, punctuation marks and emoticons may need to be preserved. Tweets preprocessing may involve converting many types of emoticons into tags that express their sentiment (i.e. **:(** $\rightarrow$ **unhappy, sad**) [24]. Therefore, emoticons play an essential role in tweets analysis as they serve the purpose of detecting emotion. Moreover, tweets are often too short in length, which makes the sentiment analysis process difficult.

Generally, as far as the Count Vectorizer technique is concerned, the models with the highest accuracy are the two Neural Networks and the Logistic Regression classifier. Regarding the TF-IDF Vectorizer technique, the models with the highest accuracy are the Neural Network II and the Logistic Regression classifier. The Neural Network I and the Linear SVM only performed good accuracy rates in some datasets.

Finally, comparing all the mentioned models, those with the best accuracy rates are the Neural Network II and the Logistic Regression classifiers using the TF-IDF Vectorizer technique. These models achieve accuracies between 0,78 to 0,88, excluding the two Twitter datasets.

5 Conclusion

In this work, several classification algorithms were implemented and evaluated on sentiment-labeled datasets. The current state-of-the-art accuracy on IMDb [24], scores 97.42%, using document embeddings trained with cosine similarity. The scope of this work is different. The goal is not only to achieve good accuracy rates but also to get good generalization results on various datasets, by using traditional machine learning techniques. Logistic Regression and the second implementation of the Neural Network worked best on the IMDb dataset. More specifically, an accuracy score of 0,8888 was achieved on this dataset. Furthermore, the impact of constructing the feature vectors, on the performance of sentiment analysis classifiers, was analyzed. The Neural Network II with the second approach which takes the top 50,000 words or bi-grams to construct the feature vectors, generalizes better that the Neural Network I on every dataset. Common preprocessing techniques were used such as lowercasing, lemmatization, stopwords and non-word removal. All techniques provided significant improvements to the classifiers performances. Some of the techniques simply removed noise from the data, while others increased the importance of specific words or combinations of words. None of the classifiers performed well in the Twitter datasets. These moderate percentages may also be due to the misspellings occurring in data [25]. It is obvious that analyzing social media data is hard, as many of them are composed of misspelled words and special words or expressions. These are difficult for researchers to understand, as they are not used in everyday speech.

However, sentiment analysis is not only useful for classification problems. An important perspective for future work would be to use sentiment analysis as a recommendation tool. Finally, datasets with discrete rating scores of each review (e.g. from 1 to 5) could potentially lead to better performance of the models used.

References

1. Angiani, G., et al.: A comparison between preprocessing techniques for sentiment analysis in Twitter. In: KDWeb (2016)
2. Cambria, E., Olsher, D., Rajagopal, D.: SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
3. Younis, E.M.G.: Sentiment analysis and text mining for social media microblogs using open source tools: an empirical study. Int. J. Comput. Appl. **112**(5), 44–48 (2015)
4. Lopez, B., Minh, A.N., Xavier, S.: IMDb sentiment analysis. COMP 551 - Group 17 (2019)
5. Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.): ICONIP 2016. LNCS, vol. 9950. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46681-1
6. Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.): ECIR 2018. LNCS, vol. 10772. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76941-7
7. Camacho-Collados, J., Pilehvar, M.T.: On the role of text preprocessing in neural network architectures: an evaluation study on text categorization and sentiment analysis. arXiv preprint arXiv:1707.01780 (2017)
8. Camacho-Collados, J., Pilehvar, M.T.: From word to sense embeddings: a survey on vector representations of meaning. J. Artif. Intell. Res. **63**, 743–788 (2018)

9. Wagh, B., Shinde, J.V., Kale, P.A.: A Twitter sentiment analysis using NLTK and machine learning techniques. Int. J. Emerg. Res. Manag. Technol. **6**(12), 37–44 (2017)
10. Martineau, J.C., Finin, T.: Delta TFIDF: an improved feature space for sentiment analysis. In: Third International AAAI Conference on Weblogs and Social Media (2009)
11. Toman, M., Tesar, R., Jezek, K.: Influence of word normalization on text classification. Proc. InSciT **4**, 354–358 (2006)
12. Aisopos, F., Papadakis, G., Varvarigou, T.: Sentiment analysis of social media content using N-Gram graphs. In: Proceedings of the 3rd ACM SIGMM International Workshop on Social Media (2011)
13. Avinash, M., Sivasankar, E.: A study of feature extraction techniques for sentiment analysis. In: Abraham, A., Dutta, P., Mandal, J., Bhattacharya, A., Dutta, S. (eds.) Emerging Technologies in Data Mining and Information Security, vol. 814, pp. 475–486. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1501-5_41
14. Tarimer, İ., Çoban, A., Kocaman, A.E.: Sentiment analysis on IMDB movie comments and Twitter data by machine learning and vector space techniques. arXiv preprint arXiv:1903.11983 (2019)
15. Pelaez, A., Talal, A., Mohsen, G.: Sentiment analysis of IMDb movie. Mach. Learn. **198**(536) (2015)
16. Oswal, N.: Predicting rainfall using machine learning techniques. arXiv preprint arXiv:1910.13827 (2019)
17. Nicapotato: Women's E-Commerce Clothing Reviews. Kaggle, 3 February 2018. www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews
18. Stanford Network Analysis Project: Amazon Fine Food Reviews. Kaggle, 1 May 2017. www.kaggle.com/snap/amazon-fine-food-reviews
19. Datafiniti: Hotel Reviews. Kaggle, 24 June 2019. www.kaggle.com/datafiniti/hotel-reviews
20. Datafiniti: Consumer Reviews of Amazon Products. Kaggle, 20 May 2019. www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products
21. Youben: Twitter Sentiment Analysis. Kaggle, 21 October 2018. www.kaggle.com/youben/twitter-sentiment-analysis/data
22. mistryjimit26: Twitter Sentiment Analysis Basic. Kaggle, 21 September 2018. www.kaggle.com/mistryjimit26/twitter-sentiment-analysis-basic/data
23. Kibriya, A.M., Frank, E., Pfahringer, B., Holmes, G.: Multinomial Naive Bayes for text categorization revisited. In: Webb, G.I., Yu, X. (eds.) AI 2004. LNCS (LNAI), vol. 3339, pp. 488–499. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30549-1_43
24. Thongtan, T., Phienthrakul, T.: Sentiment classification using document embeddings trained with cosine similarity. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (2019)
25. Agarwal, A., et al.: Sentiment analysis of Twitter data. In: Proceedings of the Workshop on Language in Social Media (LSM 2011) (2011)