A Hybrid Learning Approach for Text Classification Using Natural Language Processing

Chapter · July 2022

DOI: 10.1007/978-3-031-07969-6_32

3 authors:

Iman El Mir
Université Hassan 1er

Said El Kafhali
Université Hassan 1er

Abdelkrim Haqiq
Université Hassan 1er

A Hybrid Learning Approach for Text Classification Using Natural Language Processing

Iman El Mir[1], Said El Kafhali[2(✉)], and Abdelkrim Haqiq[2]

[1] Hassan First University of Settat, Institute of Sports Sciences, Computer, Networks, Modeling, and Mobility Laboratory (IR2M), B.P. 539, 26000 Settat, Morocco
iman.elmir@uhp.ac.ma

[2] Hassan First University of Settat, Faculty of Sciences and Techniques, Computer, Networks, Modeling, and Mobility Laboratory (IR2M), B.P. 577, 26000 Settat, Morocco
{said.elkafhali,abdelkrim.haqiq}@uhp.ac.ma

Abstract. Text classification and categorization is a hot topic that involves assigning tags or categories to a text based on its content. It is one of the important tasks of automatic natural language processing (NLP) in many applications such as topic tagging, sentiment analysis, intent detection, spam filtering, and email routing. Machine learning text classification can support businesses to automatically analyze and structure their textual documents promptly and inexpensively, to automate processes and improve data-driven decisions. In this article, we propose a new algorithm to classify textual documents using a hybrid approach that combines a set of given algorithms, using the best for each class. These documents can be classified into a set of possible class labels given a priori. Two machine learning algorithms are used to evaluate our proposed approach: Naive Bayesian (NB) and Logistic Regression (LR). The obtained results showed that the proposed hybrid algorithm is more efficient than NB and LR algorithms with an accuracy of 91.86%.

Keywords: Text classification · Machine learning · Natural language processing · Naive Bayesian · Logistic Regression

1 Introduction

The classification and categorization of textual documents is the activity of the classic problem in NLP, which consists in automatically classifying documentary resources, generally, from a corpus [1]. This classification can take an infinite number of forms. We can cite the classification by genre, by theme, or by opinion. The task of classification is performed with specific algorithms, implemented by information processing systems. It is a task of automating a classification process, which most often involves numerical methods. The document classification

M. Lazaar et al. (Eds.): BDIoT 2021, LNNS 489, pp. 428–439, 2022.
https://doi.org/10.1007/978-3-031-07969-6_32

activity is essential in many economic fields: it makes it possible to organize documentary corpus, sort them, and help to exploit them in different sectors such as administration, aeronautics, research on the internet, science, etc. [2]. The text classification algorithms have already been implemented for commercial and academic purposes. They can be mainly arranged in the following phases, namely, feature extraction, dimension reductions, classifier selection, and evaluation. Among the techniques of text classification, we find term weighting methods that conceive appropriate weights to the explicit terms to improve the performance of text classification.

The application and use of NLP help in rapid recognition, text analysis, language translation, natural language understanding, natural language generation, as well as other functions [3]. Usually, it includes two methods, such as statistical NLP and semantic NLP. Machine language is the basis for statistical NLP (including deep neural networks), which has increased the level of accuracy in text classification. Text classification is one of the most typical tasks of supervised machine learning. Assigning categories of text documents, which can be a library book, web page, gallery, media articles, etc. have many applications like sentiment analysis, intent detection, spam filtering, email routing, topic tagging, etc. In the context of text classification, machine learning is of particular importance that estimates an unknown dependence between data and output of the considered system based on available samples for proper classification. A system classification learns to classify new features into predefined discrete issue classes.

One of the most important applications of text classification is sentiment analysis that analyzes people's opinions and feelings, attitudes, and emotions [4]. It is one of the most active research areas in NLP and it is also widely studied in data mining. This research has extended outside of science to management science and social science. The growing importance of sentiment analysis coincides with the growth of social media such as Twitter and social media. For the first time in human history, we have a considerable volume of opinions in digital form for analysis [5]. Sentiment analysis systems are applied in almost all areas of business and society because opinions are central to almost all human activities and are key influencers of our behavior. Our beliefs and perceptions of reality, and the choices we make, are largely conditioned by how others see and value the world. For this reason, when we have to make a decision, we often seek the opinions of others. This is true not only for individuals but also for organizations. Sentiment analysis can determine tone, emotional behaviors, and patterns in documents to assess whether the opinion expressed on a topic is positive or negative using machine learning techniques [6].

There is a generous number of machine learning techniques that are used in the literature for text classification. However, all of them do not have identical accuracy, that is, one may have low accuracy, while others may have higher accuracy than the other. In this paper, the text classification is done using two classical machine learning algorithms that have been implemented on the AG's new topic classification dataset. The two popular classification machine learning used are logistic regression classifier and naive Bayesian classifier. The two used algorithms are playing a significant role in text classification and obtained the best results in different existing research works [7]. In addition to the two algorithms, we have proposed a hybrid algorithm for text classification and compare its performance with the above-mentioned algorithms. The main task here is to assign unlabeled new text documents to predefined classes. The main objective is to present an approach to determine how textual documents can be classified using a hybrid algorithm, based on a simple concept; choose the best algorithm for each class. This approach can improve the accuracy of the classification results.

The remainder of this paper is organized as follows. Section 2 presents some related works in the field of text classification. The text classification process is described in Sect. 3. The classifier models and the proposed hybrid algorithm are presented in Sect. 4. Section 5 discusses the evaluation of the used methods as well as our proposed algorithm for text classification. Finally, Sect. 6 concludes the paper.

2 Related Work

There have been important research papers on the advantages of machine learning algorithm development for text classification. In this section, we present some related works in the field of text classification. Numerous techniques, datasets, and evaluation metrics have been introduced in the literature and provided in multiple contributions. Following the text and models used for feature extraction and classification, the authors [8] proposed a taxonomy for text classification. Indeed, they compared the different existing techniques and developed a detailed study of the technical developments and datasets that enable prediction tests. The authors [9] examine the text classification techniques and deal with different term weighting to compare the different classification techniques using machine learning algorithms. The authors [10] presented the structure and technical implementations of text classification systems in terms of the pipeline. It is mandatory to choose the best classification algorithm for document categorization. The pipeline is divided into two sections predicting the test set and evaluating the model.

Some researchers apply feature selection models for text classification including filter, wrapper, embedded, and hybrid. The authors [11] presented the primary feature selection techniques for text classification. They introduced the Nearest Neighbor method, Naive Bayes, Support Vector Machine, Decision Tree, and Neural Networks as the most common text classifiers to discuss document representation schemes, and similarity between documents. To enhance the classification accuracy, several studies have been done using optimization algorithms. For example, the authors [12] proposed a novel firefly algorithm-based feature selection for Arabic text classification. Consequently, they suggested a new Arabic Text Categorizer system; to validate the proposed feature selection method for enhancing Arabic Text Classification accuracy; they carried out some experiments using real datasets and compare the state-of-the-art techniques with the proposed method. Due to the problem of small samples and high dimensionality for text classification, the evaluation of feature selection methods is still complicated because it involves multiple criteria. To develop a better evaluation method, several criteria must be taken into account. The authors [13] investigated multiple criteria decision-making-based methods to evaluate feature selection methods for text classification with small sample datasets according to classification performance, stability and efficiency.

In other words, the authors [7] presented a BBC news text classification model based on machine learning algorithms. Indeed, they proposed logistic regression, random forest, and K-nearest neighbor algorithms that describe each aspect of the model in detail providing the evaluation metrics. The accuracy remains the most important parameter that must be taken into consideration when machine learning algorithms are implemented on a particular data set. The obtained experimental results prove the efficiency of the BBC news text classification model according to the algorithms tested on the data set. The authors [1] studied some deep learning models for Arabic news classification. To avoid the need for the pre-processing phase, they applied deep learning hence to produce a highly accurate and robust classifier for Arabic news articles. Authors in [14], have proposed to classify text documents within a certain hierarchy. However, deep neural models remain difficult to apply and define the levels of documents hierarchically because they need a large amount of training data. The authors [15] proposed a novel Text Classification Framework called SS3, whose aim is to provide support for incremental classification, early classification for simple and effective early depression detection.

Long Short-Term Memory (LSTM) is the most common architecture, which is designed to catch long term dependencies. The LSTM network is applied in each language to model the documents [16]. The classification is performed by using a hierarchical attention mechanism, where the sentence-level attention model determines which sentences of a document are more significant for giving the overall sentiment. While the word-level attention model elaborates which words in each sentence are decisive. Comparing Bidirectional Encoder Representations with Transformer (BERT) Bidirectional Encoder Representations with directional models like RNN and LSTM, it is noted that these latter sequentially consider each input while BERT is nondirectional and read the whole sentence as the input instead of sequential processing. BERT remains more efficient in terms of results and learning speed [17]. Once pretrained, in an unsupervised manner, it has its linguistic representation. It is then possible, based on this initial representation, to customize it for a particular task. It can be trained in an incremental mode in a supervised fashion to specialize the model quickly and with little data. As, it can work in a multimodel way, taking as input data of different types like images, text. Finally, the authors [18] used the deep learning technique to deal with the sentiment classification. Indeed, they proposed convolutional neural networks for sentiment classification. They tackle the sentiment analysis that classifies a relatively longer text into one of the sentiment categories.

3 Text Classification Process

The objective of the so-called classical Machine Learning is to give a machine the ability to learn to solve a problem without having to explicitly program each rule. The idea of machine learning is therefore to solve problems by modeling behavior through data-driven learning. However, before being able to model a problem through a machine-learning algorithm, it is often necessary to perform several transformations on the data. These transformations, which are done manually, are dictated by the business problem we are trying to solve, and by the choice of the algorithm used. This data processing, called feature engineering, is often very time consuming and may require business expertise to be relevant. We can consider these data transformations as the construction of a representation of the problem that can be easily understood and interpreted by the machine learning algorithm.

In text classification process, at the start, text documents are read from the collection, then preprocessing like tokenization, stop word elimination, stemming. After that, text representation in a form so that learning algorithms can be applied. Finally, important features are selected from the feature vector to remove the irrelevant features and minimize the dimensionality of feature space. To do that, we should follow the steps as illustrated in the flow diagram of Fig. 1.

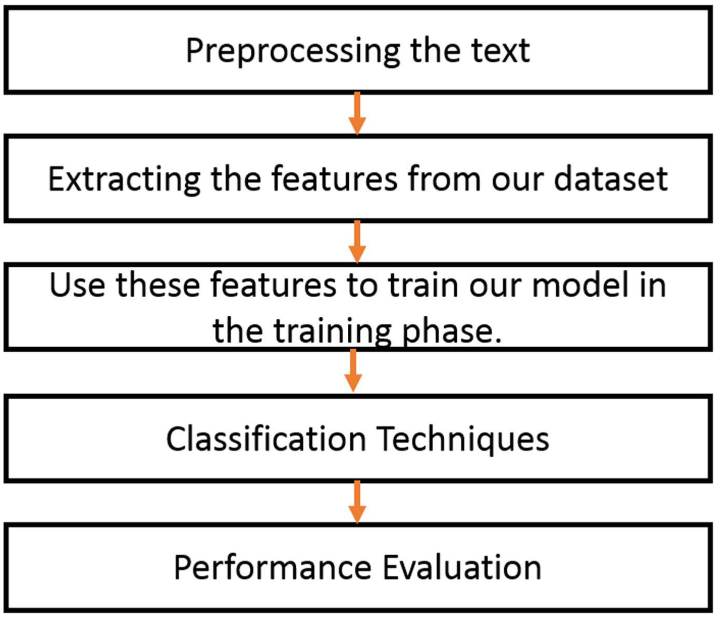

Fig. 1. Flow diagram of the proposed approach

3.1 Preprocessing

Tokenization. Tokenization is the first pre-processing stride of natural language processing and corpus generation. It is the process of replacing the meaningful sentence into individual words with space as the delimiter and it retains all valuable information. Each word is known as tokens. These tokens are the key elements of NLP.

Stop Words Elimination. Stop words are a part of the natural language that does not have so much meaning in a retrieval system. The reason that stop words should be removed from a text is that they make the text look heavier and less important for analysts. Removing stop words reduces the dimensionality of the term space. The most common words are in text documents are prepositions, articles, and pronouns, etc. that do not provide the meaning of the documents. These words are treated as stop words.

Stemming. The stemming technique is used to determine the root/stem of a word. It converts the words into their stems, which incorporates much of the language-dependent linguistic knowledge.

Feature Selection. Is an essential technique in dimensionality reduction to extract the most important features. It is an operation of converting text into a collection of features as a real number next to a vector which will be used as the input of classifier [19]. The operation of extracting a feature from the text, use the bag of words that changes all text in a dictionary consists of all words that appear in all texts. It creates a collection of entities as real numbers inside a vector for each text where the value of each entity inside the vector will be founded on the rate of occurrence of each word enumerated in the text.

After doing the preprocessing, we should go to the feature selection step. In this step, the primary task is to minimize the feature space dimensionality. The subset entities available in the dataset are the selected keywords. The selected features receive the highest scores based on a function that measures the importance of the feature to the text classification task. The used functions to measure the importance of the feature are most significant. The simple and effective function is the term frequency of a term, which is only those terms that appear in the highest numbers in a document are kept.

3.2 Text Representation

There are many techniques for text representation, including the Term Frequency, Inverse Document Frequency (TF-IDF) technique and others of its modifications by applying dimensionality reduction techniques such as Latent Semantic Analysis (LSA) and linear discriminant analysis (LDA).

In this article, we have used TF-IDF [20] that is a numerical statistic whether reveals that a word is how important to a text document in a collection. TF-IDF is frequently used as a weighting factor in information retrieval and text mining. The value of TF-IDF increases proportionally to the number of times a word appears in the text document but is counteracted by the frequency of the word in the corpus. This can help to supervision the fact that a number of words are mostly more common than others. TF-IDF can be successfully used for stop-words filtering in various subject fields including text classification and summarization. TF-IDF is the product of two statistics that are term frequency and inverse document frequency. To more distinguish them, the number of times each term occurs in each text document is counted and sums them together.

Term Frequency (TF) is defined as the number of times a term occurs in a document:

$$TF(word, document) = 0.5 + \frac{(0.5 \times f(word, document))}{\text{Maximum occurrences of words}} \tag{1}$$

An Inverse Document Frequency (IDF) is a statistical weight used for measuring the significance of a term in a text document collection. IDF feature is incorporated which minimizes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely:

$$IDF(word, document) = \beta \times \log\left(\frac{\text{Total number of documents}}{\text{Number of documents in which word appears}}\right) \tag{2}$$

with β is the number of times word appears in document.

Then the TF-IDF is calculated as:

$$TF - IDF = TF(word, document) \times IDF(word, document) \tag{3}$$

4 Classification Techniques

Several techniques are utilized to classify the text documents like NB, support vector machine (SVM), k-nearest neighbor (KNN), and LR. In this paper, we have used NB and LR algorithms, which are presented in the next sections.

4.1 Naive Bayesian Algorithm

NB is a classification machine learning technique based on an assumption of independence between predictors or what is called Bayes' formula. An NB classifier assumes that the presence of a particular characteristic in a class is not related to the presence of another characteristic [9]. The BN classifier works by taking a text document to classify and calculate the probability that the text document falls into each of the categories with which the system has trained. It generates a unique training category with the highest probability of containing the document to be classified and produces a probability for each possible class to identify several categories to which a document can belong. To categorize and classify text documents by NB, we use the following equation:

$$P(class|document) = \frac{P(class) \times P(document|class)}{P(document)} \tag{4}$$

The probabilities used in the previous equation are defined as follows:

- $P(class|document)$: is the probability that a given text document belongs to a given class.
- $P(document)$: is the probability of a document. It is a constant, we can ignore it.
- $P(class)$: is the probability of a class that is calculated from the number of text documents in the category divided by the text documents number in all categories. Let D_c represent the number of documents in a given class and D_t represent the number of documents we have total. We can calculate the probability of a class by:

$$P(class) = \frac{P(D_c)}{P(D_t)} \tag{5}$$

- $P(document|class)$: is the probability of a text document given class.

Text documents can be represented by a sets of words, hence we have the following equations:

$$P(document|class) = \prod_i P(word_i|class) \tag{6}$$

$$P(document|class) = P(class) \times \prod_i P(word_i|class) \tag{7}$$

With $P(word_i|class)$ is the probability that a given word occurs in all text documents of a given class.

4.2 Logistic Regression Algorithm

LR comes under the supervised learning classification algorithm. In recent years, this algorithm has achieved importance and its use has increased extensively in different field. LR is a classification machine learning algorithm that estimates discrete values (binary values such as 0/1, yes/no, true/false) based on a given set of independent variables. It predicts the probability of an event occurring. RL is a discriminant method that consists in calculating $P(y|x)$ by discriminating between the different possible values of the class y based on the given input class x that is as shown below:

$$P(c|x) = \sum_{i=1}^{N} w_i f_i(x) \tag{8}$$

We cannot calculate the value $P(y|x)$ directly by using the Eq. (8) because it will result in a value from $-\infty$ to $+\infty$ which means it will not result from an output between value 0 and 1.

To get a value of output that between the value 0 and 1, the following equation is used

$$P(c|x) = \frac{exp(\sum_{i=1}^{N} w_i f_i(x))}{\sum_c exp(\sum_{i=1}^{N} w_i f_i(x))} \tag{9}$$

N specify the number of features.

It is usual in language processing to employ binary-valued features. The features are not only a property of the observation x, but also are a property for both the observation x and the candidate output class c [21]. So, instead of $f_i(x)$, we use the function $f_i(c, x)$ that consists to whereby feature i of class c is assigned to the given entry of x. Therefore, the equation to calculate the probability of y being of class c given x becomes

$$P(c|x) = \frac{exp(\sum_{i=1}^{N} w_i f_i(c, x))}{\sum_{j \in c} exp(\sum_{i=1}^{N} w_i f_i(j, x))} \tag{10}$$

4.3 The Proposed Hybrid Approach

This approach combines the two algorithms above. In this approach, we will choose the best algorithms for each class, which means that every class is predicted by one and only one algorithm. The algorithms must be trained apriori over the dataset. Our proposed algorithm can be presented as in Algorithm 1.

Require: Decision_table: tab_decision
Ensure: Accuracy_table: tab_accuracy
1. i=0
2. **while** algo ∈ algorithmes **do**
3. tab_accuracy←accuracy(algo)
4. **end while**
5. **for** row ∈ test **do**
6. Initialise tab_decision by 0 for each class
7. **for** algo ∈ algorithms **do**
8. class_predicted←predict algo(row)
9. tab_decision(class_predicted)+=tab_accuracy(algo)
10. **end for**
11. classe_predicted=key(max((tab_decision))
12. **if** classe_predicted is True **then**
13. i+=1
14. **end if**
15. **end for**
16. Accuracy= i/size(test)
17. **Return** Accuracy

Algorithm 1: Hybrid algorithm

5 Performance Evaluation

5.1 Classification Dataset

We have used the AG's news topic classification dataset [22] that is a set of over one million news articles (war, sport, business, science, and technology). News articles have been gathered from over 2000 news sources in over a year of activity. The AG's news topic classification dataset is provided by the university community for research goals in data mining, data streaming, data compression, and every other noncommercial activity.

The AG's news topic classification+ dataset is constructed by choosing the four largest classes from the original corpus in September 2015, and it is used as a text classification benchmark by Zhang *et al.* [23]. Each class contains 30,000 training samples and 1,900 testing samples. There are 120,000 articles for the training set and 7,600 articles for the test.

5.2 Results and Discussion

We use the accuracy parameter to evaluate the efficiency of our proposed technique. Accuracy can be defined as the ratio between the number of correctly categorized texts and the entire text. Table 1 presents a confusion matrix is utilized to calculate the accuracy metric.

Table 1. Confusion matrix

		Predicted class	
		Categorized positive	Categorized negative
Actual class	Categorized positive	True positive	False negative
	Categorized negative	False positive	True negative

The accuracy is defined by Eq. (11) that measures the ratio of correct predictions over the total number of instances evaluated.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

With:

- True Positive (TP) refers to the collection of text documents that are assigned correctly to the given category.
- True Negative (TN) refers to the collection of text documents that are not assigned correctly to the given category.
- False Positive (FP) refers to the collection of text documents incorrectly assigned to the given category.
- False Negative (FN) refers to the collection of text documents incorrectly not assigned to the given category.

Table 2 shows the accuracies for each algorithm and the hybrid algorithm. We found that the Hybrid classifier outperformed NB and LR.

Table 2. Accuracies for each algorithm

Algorithm	Accuracy
Naive Bayesian algorithm (NB)	90.27%
Logistic Regression algorithm (LR)	91.40%
Hybrid algorithm	91.86%

6 Conclusion

We have presented a hybrid approach for text classification that combines a set of given algorithms, using the best for each class. Feature reduction and selection methods are used in combination with NB and LR learning algorithms to increase the accuracy of the classifier. The proposed approach has proved its efficiency by outperforming the given algorithms in our case. The future direction of this work is to increase the volume of the dataset and use other algorithms of machine learning and deep learning, with different feature selection methods like doc2vec. We also try to experiment with more hybrid algorithms to obtain high benefits from machine learning techniques and to reach better classification results.

References

1. Elnagar, A., Al-Debsi, R., Einea, O.: Arabic text classification using deep learning models. Inf. Process. Manag. **57**(1), 102121 (2020)
2. Hartmann, J., Huppertz, J., Schamp, C., Heitmann, M.: Comparing automated text classification methods. Int. J. Res. Mark. **36**(1), 20–38 (2019)
3. Liu, H., Yin, Q., Wang, W.Y.: Towards explainable NLP: a generative explanation framework for text classification. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 5570–5581 (2019)
4. Yadav, A., Vishwakarma, D.K.: Sentiment analysis using deep learning architectures: a review. Artif. Intell. Rev. **53**(6), 4335–4385 (2019). https://doi.org/10.1007/s10462-019-09794-5
5. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. J. **5**(4), 1093–1113 (2014)
6. Jain, A.P., Dandannavar, P.: Application of machine learning techniques to sentiment analysis. In: 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 628–632. IEEE (2016)
7. Shah, K., Patel, H., Sanghvi, D., Shah, M.: A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Hum. Res. **5**(1), 1–16 (2020). https://doi.org/10.1007/s41133-020-00032-0

8. Li, Q., et al.: A survey on text classification: from shallow to deep learning. ACM Comput. Surv. **37**(4), 1–35 (2020)
9. Kadhim, A.I.: Survey on supervised machine learning techniques for automatic text classification. Artif. Intell. Rev. **52**(1), 273–292 (2019). https://doi.org/10.1007/s10462-018-09677-1
10. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: a survey. Information **10**(4), 150 (2019)
11. Deng, X., Li, Y., Weng, J., Zhang, J.: Feature selection for text classification: a review. Multimedia Tools Appl. **78**(3), 3797–3816 (2018). https://doi.org/10.1007/s11042-018-6083-5
12. Marie-Sainte, S.L., Alalyani, N.: Firefly algorithm based feature selection for Arabic text classification. J. King Saud Univ.-Comput. Inf. Sci. **32**(3), 320–328 (2020)
13. Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., Alsaadi, F.E.: Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. Appl. Soft Comput. **86**, 105836 (2020)
14. Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised hierarchical text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6826-6833, AAAI Press, Palo Alto, California USA (2019)
15. Burdisso, S.G., Errecalde, M., Montes-y-Gómez, M.: A text classification framework for simple and effective early depression detection over social media streams. Expert Syst. Appl. **133**, 182–197 (2019)
16. Sachan, D. S., Zaheer, M., Salakhutdinov, R.: Revisiting lstm networks for semi-supervised text classification via mixed objective function. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6940–6948, AAAI Press, Palo Alto, California USA (2019)
17. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) CCL 2019. LNCS (LNAI), vol. 11856, pp. 194–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16
18. Kim, H., Jeong, Y.S.: Sentiment classification using convolutional neural networks. Appl. Sci. **9**(11), 2347 (2019)
19. Dzisevič, R., Šešok, D.: Text classification using different feature extraction approaches. In: 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), pp. 1–4. IEEE (2019)
20. Christian, H., Agus, M.P., Suhartono, D.: Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). ComTech Comput. Math. Eng. Appl. **7**(4), 285–294 (2016)
21. Indra, S. T., Wikarsa, L., Turang, R.: Using logistic regression method to classify tweets into the selected topics. In: 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 385–390. IEEE (2016)
22. http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
23. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 649–657 (2015)