Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 5, No. 3, August 2020, Pp. 235-242

Document preprocessing with TF-IDF to improve the polarity classification performance of unstructured sentiment analysis

Farrikh Alzami*[1], Erika Devi Udayanti[2], Dwi Puji Prabowo[3], Rama Aria Megantara[4]
Universitas Dian Nuswantoro, Semarang, Indonesia[1, 2, 3,4]

Article Info
Keywords:
Unstructured Sentiment Analysis, Polarity, TF-IDF, Classification

Article history:
Received 18 April 2020
Revised 15 July 2020
Accepted 25 July 2020
Published 31 August 2020

Cite:
Alzami, F., Udayanti, E., Prabowo, D., & Megantara, R. (2020). Document preprocessing with TF-IDF to improve the polarity classification performance of unstructured sentiment analysis. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 5(3). doi:https://doi.org/10.22219/kinetik.v5i3.1066

*Corresponding author.
Farrikh Alzami
E-mail address:
alzami@dsn.dinus.ac.id

Abstract
Sentiment analysis in terms of polarity classification is very important in everyday life, with the existence of polarity, many people can find out whether the respected document has positive or negative sentiment so that it can help in choosing and making decisions. Sentiment analysis usually done manually. Therefore, an automatic sentiment analysis classification process is needed. However, it is rare to find studies that discuss extraction features and which learning models are suitable for unstructured sentiment analysis types with the Amazon food review case. This research explores some extraction features such as Word Bags, TF-IDF, Word2Vector, as well as a combination of TF-IDF and Word2Vector with several machine learning models such as Random Forest, SVM, KNN and Naïve Bayes to find out a combination of feature extraction and learning models that can help add variety to the analysis of polarity sentiments. By assisting with document preparation such as html tags and punctuation and special characters, using snowball stemming, TF-IDF results obtained with SVM are suitable for obtaining a polarity classification in unstructured sentiment analysis for the case of Amazon food review with a performance result of 87,3 percent.

1. Introduction

Sentiment analysis using text analysis, natural language processing (NLP), and computation techniques to automate the extraction or classification of a sentiment in sentiment reviews [1]. Analysis of an opinion or sentiment is very important in many fields, such as: e-health [2], political [3], financial [4][5], tourism [6] information on consumer needs [7], applications [8], books [9], social media [10] and websites [11]. Sentiment analysis is also an area that is often developed in decision making [12]. For e-commerce, sentiment analysis is very important, this is because customers usually want good quality products at the lowest possible price but cannot check directly, so reading reviews from other customers is the most appropriate way to decide whether to buy the product or not. Therefore sentiment analysis proves important in understanding the popularity of the product [13]. Sentiment analysis for product reviews, commonly called opinion mining, refers to the process of automatically analyzing subjective commentary texts originating from emotional tendencies [14].

The main purpose of sentiment analysis is to analyze the review and calculate the score of the sentiment. The reviews obtained can be grouped into positive, negative or neutral reviews, this is referred to as sentiment polarity [15]. Review sentiments that currently appear can be grouped into three parts [16], including: (1) structured sentiment, can be found in official reviews, such as in research reviews or book review reviews, this occurs because reviews are conducted by professionals; (2) semi-structured sentiment, usually found in the pros and cons discussion; (3) Unstructured sentiment, this can be found in informal and free writing that does not follow the correct writing rules [17].

To determine a review of a sentence, especially in a document, feature extraction is needed to get feature vectors. From these feature vectors, they will then be trained to use classification learning models to obtain the results of the polarity of the review. Currently, there are several methods for making feature vectors for document data types, including: Bag of Words (BoW), Term Frequency and Inverse Document Frequency (TF-IDF), Word 2 Vector, and the combination of TF-IDF with Word 2 Vector.

Amazon food reviews dataset [18] is data that contains 568,454 reviews of foods from Amazon online stores. Amazon food reviews dataset is quite often used for sentiment analysis, but it is quite rare that it explores the relationship between feature extraction and machine learning.

To improve the performance of the sentiment analysis model, this study proposes an exploratory approach to the relationship between feature extraction and learning models to get a better polarity classification. Thus, main contribution of this research are: 1) Exploring preprocessing documents to improve the quality of documents to be

processed; 2) Explore and find out the factors that influence feature extraction with learning models to improve the performance of sentiment analysis.

In general, these manuscripts are written in the following order: section 2 describe the research method, the results and discussion are listed in section 3, concluding and future research listed in section 4.

2. Research Method

This study uses the following stages: 1) document preprocessing; 2) feature extraction in document 3) application of features extracted by some machine learning such as random forest, SVM, Multinomial naïve Bayes and KNN to find out which feature extraction is better. 3-fold cross validation was also used in this study to reduce random effects.

2.1 Document preprocessing stage

In this Amazon food reviews document preprocessing, the first step is to clear the data with several stages, including: 1) on Amazon food reviews, the label is a review score with a range of numbers from 1 to 5, then a score of 1 or 2 is transformed into a negative review polarity, 3 is the neutral review polarity and 4 or 5 is considered the positive review polarity. For this study, only positive and negative polarity were used, while neutral polarity was not included in this study; 2) delete the duplicate data that is on the amazon food reviews dataset. This duplicate data is found by searching for users who make multiple reviews at the same time. So that from the original data which amounted to 525814 records to 364171 records; 3) grouping data that has positive and negative scores, so that a positive score of 307061 records and negative 57110 records is found; 4) here the most important steps are: a) delete the tags listed in records such as html tags, b) delete punctuation marks and special characters, c) only consider English (because amazon food reviews are mostly in English), d) delete alpha-numeric, e) change the writing to lowercase (small writing) f) use snowball stemming (a small string processing language designed to make the stemming algorithm for use in information retrieval) [19]. Keep in mind, at this stage, stop words are not removed because this review is included in unstructured sentiment and stop words can often improve the performance of feature extraction for unstructured sentiment.

2.2 Feature extraction stage

2.2.1 Bag of Words

Bag of Words (BoW) can be seen as a machine that receives input of a document and outputs a table containing the number of word frequencies available for each document. For example, there are three documents with the following sentence: 1) I like cheese; 2) I am allergic to cheese and milk; 3) I like milk. From those 3 sentences, we can obtain the BoW as follows Table 1.

Table 1. BoW Example

	I	Like	Cheese	Allergic	And	Milk
Doc. 1	1	1	1	-	-	-
Doc. 2	1	-	1	1	1	1
Doc. 3	1	1	-	-	-	1

The drawbacks of BoW are: BOW does not consider semantic meaning. Example: delicious and tasty have the same meaning but BOW considers it separate.

2.2.2 TF-IDF

For specific documents, Term Frequency (TF) determines how important a word is seen from how often the words appear in a document. We can say that TF is the output of BoW. In TF-IDF, the second component is inverse document frequency (IDF). In IDF, a word is considered as important in a document if the word does not appear very often in other documents. This can be calculated as follows Equation 1.

$$idf(words) = log \frac{number\ of\ documents}{the\ number\ of\ documents\ containing\ the\ word} \quad (1)$$

For example, it can be seen in Table 2 using the BoW example.

Table 2. TF-IDF Example

	I	Like	Cheese	Allergic	And	Milk
Doc. 1	0	**0.18**	**0.18**	-	-	-
Doc. 2	0	-	0.18	**0.48**	**0.48**	0.18
Doc. 3	0	**0.18**	-	-	-	**0.18**

From Table 2, we can see that in document 1 the highlight is ‘like’ ’cheese’, in document 2 it is ‘allergic’ ‘and’, in document 3 it is ‘like’ ‘milk’. Please note, the words 'and' in general will be deleted using stop words before feature extraction is performed.

The drawback of TF-IDF is that it does not capture position in text, semantics, and co-occurrence in various documents.

In document processing, text representation schemes usually use vector space models (VSM) which are often used for word weighting. The results received from VSM are relevant documents. VSM used here uses keywords or phrases, commonly known as unigrams, bigrams, trigrams and n-grams [20]. For simplicity's sake, the N-gram is a sequence of N words. For example, there is a sentence: "This food is not very tasty", so if you make n-gram, you get the following:

1. Unigram: ‘this’, ‘food’, ‘is’, ‘not’, ‘very’, ‘tasty’
2. Bigram: ‘this food’, ‘food is’, ‘is not’, ‘not very’, ‘very tasty’
3. Trigram: ‘this food is’, ‘food is not’, ‘is not very’, ‘not very tasty’

The N-gram method makes a decision by comparing this value with the similarity ratio, which is defined as the identical N-gram ratio compared to the total number of N-grams. Similarity ratios can be calculated [21], follow Equation 2.

$$similarity\ ratio = \frac{\delta}{\min(\alpha, \zeta)} \tag{2}$$

Here the words$_1$ is the first word and word$_2$ is the second word used as a comparison of the n-grams character. Please note, n-grams here are used for TF-IDF purposes.

2.2.3 Word2Vector

Word2Vector basically places words in the feature space so that their location is determined by their meaning i.e. words that have the same meaning are grouped together and the distance between two words also has the same meaning. The calculation method uses cosine similarity which can be written as follows Equation 3.

$$similarity = \cos\theta = \frac{words_1 . words_2}{\|words_1\| . \|words_2\|} \tag{3}$$

The drawbacks of Word2Vec are: 1) Sub-linear relationships are not explicitly defined; 2) and have not been able to separate several pairs of opposite words, for example, "good" and "bad" are usually located very close to each other in vector space, which can limit the performance of word vectors.

2.2.4 TF-IDF and Word2Vector

The value of TF-IDF will be calculated in each word, then multiplying the value of TF-IDF with the appropriate word and then dividing the amount by the number of TF-IDF values [22].

2.3 Implementing obtained features into the learning model

Extracted features are fed into learning model such as: support vector machine, K-nearest neighbor, naïve Bayes, Random Forest.

2.3.1 Support Vector Machine (SVM)

SVM is a model that can be used for classification and regression [23]. SVM can be explained as follows: assume (x_i , y_i) is the sample point of the data attribute pair, where $x_i \in \mathbb{R}^D , y_i \in \{+1, -1\}\ dan\ i = 1, \dots, n$ It is assumed that the positive class is denoted as +1 and the negative class as -1. In SVM, optimization is needed to solve the problem with the following Equation 4.

$$minimize: \frac{1}{2}||w||^2 \tag{4}$$

Following Equation 5.

$$y_j(\omega x_j + \beta) \geq 1, \forall j \tag{5}$$

This optimization can be completed with Lagrange Multipliers as follows Equation 6.

$$L(\omega, \beta, \gamma) = \frac{1}{2}||\omega||^2 - \sum_{j=1}^{n} \gamma_j [\gamma_j(\omega, x_j + \beta) - 1] \quad (6)$$

With $\gamma_j \geq 0, j = 1,2, \dots n$. Thus, the solution can be found as follows Equation 7.

$$max: L(\gamma) = \sum_{j=1}^{n} \gamma_j - \frac{1}{2} \sum_{j,k=1}^{n} \gamma_j \gamma_k y_j y_k x_j x_k \quad (7)$$

Following Equation 8.

$$\sum_{j=1}^{n} \gamma_j y_j = 0 \ and \ \gamma_j \geq 0, j = 1, \dots, n \quad (8)$$

2.3.2 Naïve Bayes (NB)

Naïve Bayes is a learning method that uses conditional probabilities as a basis [24]. Assume that the Y label is a random Boolean value, X_i is also a random Boolean value, where $i = 1, \dots, n$. So Bayes's theory can be represented as follows Equation 9.

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(X_1 \dots X_n | Y = y_k) P(Y = y_k)}{\sum_j P(X_1 \dots X_n | Y = y_j) P(Y = y_j)} \quad (9)$$

Where y_k is a possible value to k on Y and $X_1, X_2, \dots, X_n$ is a discrete values. Naïve Bayes estimates conditional probabilities in classes by assuming that the attribute is conditionally independent given the class label Y. The conditional independent assumption can be written as follows Equation 10.

$$P(X_1 \dots X_n | Y) = \prod_{i=1}^{n} P(X_i | Y) \quad (10)$$

Assuming that X_i where $i = 1,2, \dots n$ is a conditional independent variable for the value of Y and using Bayes theory, Equation 9 can be rewritten as Equation 11.

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) \prod_{i=1}^{n} P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_{i=1}^{n} P(X_i | Y = y_j)} \quad (11)$$

The advantages of naïve bayes are that they do not need a lot of training data, can be trained quickly, are easy to implement and do not need a lot of parameters such as SVM or neural networks. Naïve Bayes used here are multinomial Naïve Bayes. The reason for choosing the Naïve Bayes multinomial is because this learning model is commonly used in text classification and is suitable for discrete features.

2.3.3 K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) is a classification method where the new object is labeled by the nearest neighbor as K. The stages of K-NN are as follows: 1) determine the number of K (number of nearest neighbors); 2) calculate each object with sample data using Euclidean distance as follows Equation 12.

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})} \quad (12)$$

Where $X_1 = (x_{11}, x_{12}, \dots, x_{1n}), X_2 = (x_{21}, x_{22}, \dots, x_{2n})$ and $i = 1,2, \dots, n$ 3) sort the objects into groups that have the smallest Euclidean distances; 4) collect the closest neighbor categories; 5) vote on the label of the class.

2.3.4 Random Forest

Random forest consists of several trees that are combined into one model. Each tree provides a prediction of the class, and the class that is chosen is the winner. In detail, the random forest process can be described as follows: p-dimensional random vector $X = (X_1, \dots, X_p)^T$ is called a predictor variable and the random variable Y is called a response. Assume the joint distribution $P_{XY}(X, Y)$, the objective is to find the prediction function $f(X)$ to predict Y, where $L(Y, f(X))$ is the loss function used to determine the prediction function. Expectations of joint distributions on X and Y can be written as follows Equation 13.

$$E_{XY}\left(L(Y, f(X))\right) \tag{13}$$

From Equation 13, it can be seen that $L(Y, f(X))$ measures how close Y is from $f(X)$. Therefore, the penalty value of $f(X)$ is if it is far from Y. Zero-loss is usually used as a penalty by following the following rules Equation 14.

$$E_{XY}\left(L(Y, f(X))\right) \tag{14}$$

Then, from minimizing Equation 13, we get a function Equation 15.

$$f(x) = \underset{y \in \Delta}{\operatorname{argmax}} P(Y = y | X = x) \tag{15}$$

Where Δ is a collection of possible values of Y

3. Results and discussion

The Amazon food review dataset consists of 525,814 reviews; 256,059 users; 74,258 products, and timespan from October 1999 to October 2012 with the following attributes: index, id, product id, user id, profile name, helpfulness numerator (number of users who found this review helpful), helpfulness denominator (number of users indicating whether they were feel this review helps or not), score (from 1 to 5), time, summary (review summary), text (review content). The dataset is processed according to the steps in section 2 including document preprocessing, feature extraction, then using machine learning to find out which model is suitable to use.

The parameters used for this study are shown in Table 3 and parameter optimization is done by brute-force search, which is finding the best combination of parameters using accuracy measures.

Table 3. Parameter Which Used in this Study

Learning model	Parameter
Random Forest	Estimator: 100, 200,500 Max features: auto, sqrt, log2 Max depth: 4,5,6,7,8 Criterion: gini, entropy
SVM	Kernel: RBF C: 0, 500, 1000, 2000, 2500 Gamma: 10E-8, 10E-6
Naïve Bayes	Model: multinomial Alpha: 1
KNN	Distance: minkowski K: 1-40

The reason for combining scores 4 and 5 to be positive and scores 1 and 2 to negative is that there are some ambiguous reviews on scoring, for example: "*good flavor! These came securely packed ... they were fresh and delicious! I love these Twizzlers*". In that sentence, intuitively it should get a score of 5, but in a set, the score is 4. Then, because the data is quite large and some feature extraction requires a large memory (especially Word2Vector), thus, the sampling method is used in this study. First the data is sorted first first from the initial year, then reviews for two class (positive and negative) were taken as many as 5000, resulting in 10,000 records.

For the BoW extraction feature, standardization is applied for the records; For TF-IDF extraction, VSM bigram (2-gram) and standardization are applied to these records. The reason bigram is used, due to bigram is robust in part of

TF-IDF extraction [25]; for Word2Vector extraction, the layer size is 300, the minimum word to consider is 5; while for the combination of TF-IDF and Word2Vector use parameters such as TF-IDF and Word2Vector as described above. After each feature extraction is carried out, the extraction results are entered into each learning model and performance is measured. A summary of the performance can be seen in the Table 4.

Table 4. Results Using Feature Extraction and Learning Model (Where F is F-measure and MCC is Matthew Correlation Coefficient)

		RF	SVM	NB	KNN
BoW	Acc	0.818	**0.85**	0.769	0.64
	Precision	0.85	**0.85**	0.769	0.65
	Recall	0.82	**0.85**	0.769	0.64
	F	0.814	**0.85**	0.769	0.637
	MCC	0.63	**0.7**	0.54	0.295
TF-IDF	Acc	0.808	**0.873**	0.806	0.503
	Precision	0.812	**0.873**	0.808	0.434
	Recall	0.808	**0.873**	0.806	0.495
	F	0.807	**0.873**	0.806	0.345
	MCC	0.62	**0.745**	0.62	-0.03
W2V	Acc	**0.778**	0.753	-	0.7125
	Precision	**0.779**	0.757	-	0.714
	Recall	**0.777**	0.751	-	0.711
	F	**0.777**	0.751	-	0.711
	MCC	**0.557**	0.508	-	0.425
TF-IDF & W2V	Acc	**0.746**	0.727	-	0.709
	Precision	**0.746**	0.727	-	0.709
	Recall	**0.746**	0.727	-	0.709
	F	**0.746**	0.727	-	0.709
	MCC	**0.49**	0.45	-	0.417

From Table 4, it can be seen that for BoW and TF-IDF, the best performance is obtained by using SVM as a learning model compared to other models. For W2V as well as a combination of TF-IDF and W2V, the best results are obtained using Random Forest. Multinomial Naïve Bayes fail in the learning process for W2V or a combination of TF-IDF and W2V because W2V produces negative values in the making of features, while multinomial naïve Bayes cannot handle values outside of discrete. When viewed from the MCC value, TF-IDF is more suitable for the type of unstructured sentiment dataset, especially amazon food reviews, because TF-IDF has a value of 0.745 (which means strong). Another reason TF-IDF is suitable for Amazon food reviews is that the process of making features is quite fast compared to BoW and W2V. Moreover, the values of F-measure, precision, recall and Acc (accuracy) justify the findings that TF-IDF is suitable for unstructured sentiment analysis type problems.

Thus, the findings of this study are: 1) SVM using RBF is able to improve the BoW and TF-IDF performance in amazon food reviews sentiment analysis; 2) Using multinomial Naïve Bayes with Word2Vector together is not suitable for sentiment analysis

4. Conclusion

There are still few studies exploring which feature extractions are best used with learning models for sentiment analysis cases with the amazon food reviews dataset, the polarity classification for unstructured sentiment analysis is presented in this study. By using document preprocessing in the form of: removing tags listed in records such as html tags, deleting punctuation marks and special characters, only considering English, deleting alpha-numerics, changing writing to lowercase, using snowball stemming; and using the TF-IDF feature extraction can improve the SVM learning model to 0.873 (87.3 percent).

Several things that can be considered for future research are: 1) the use of feature selection to improve the performance of the polarity classification of unstructured sentiment analysis; 2) using other feature extraction such as glove and 3) using deep learning as learning model to obtain better performance.

Notation

δ : number of identical n-grams
α : number of n-gram for words$_1$
ζ : number of n-gram for words$_2$

ω: weight vector

β: bias

References

[1] Agarwal, B., Mittal, N., Bansal, P., & Garg, S. (2015). Sentiment Analysis Using Common-Sense and Context Information. *Computational Intelligence and Neuroscience*, *2015*, 1–9. https://doi.org/10.1155/2015/715730

[2] Cambria, E., Hussain, A., Durrani, T., Havasi, C., Eckl, C., & Munro, J. (2010). Sentic Computing for patient centered applications. *IEEE 10th International Conference On Signal Processing Proceedings*, 1279–1282. https://doi.org/10.1109/ICOSP.2010.5657072

[3] Ebrahimi, M., Yazdavar, A. H., & Sheth, A. (2017). Challenges of Sentiment Analysis for Dynamic Events. *IEEE Intelligent Systems*, *32*(5), 70–75. https://doi.org/10.1109/MIS.2017.3711649

[4] Xing, F. Z., Cambria, E., & Welsch, R. E. (2018). Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, *50*(1), 49–73. https://doi.org/10.1007/s10462-017-9588-9

[5] Van de Kauter, M., Breesch, D., & Hoste, V. (2015). Fine-grained analysis of explicit and implicit sentiment in financial news articles. *Expert Systems with Applications*, *42*(11), 4999–5010. https://doi.org/10.1016/j.eswa.2015.02.007

[6] Valdivia, A., Luzon, M. V., & Herrera, F. (2017). Sentiment Analysis in TripAdvisor. *IEEE Intelligent Systems*, *32*(4), 72–77. https://doi.org/10.1109/MIS.2017.3121555

[7] Vázquez, S., Muñoz-García, Ó., Campanella, I., Poch, M., Fisas, B., Bel, N., & Andreu, G. (2014). A classification of user-generated content into consumer decision journey stages. *Neural Networks*, *58*, 68–81. https://doi.org/10.1016/j.neunet.2014.05.026

[8] Thompson, J. J., Leung, B. H., Blair, M. R., & Taboada, M. (2017). Sentiment analysis of player chat messaging in the video game StarCraft 2: Extending a lexicon-based model. *Knowledge-Based Systems*, *137*, 149–162. https://doi.org/10.1016/j.knosys.2017.09.022

[9] Wang, K., Liu, X., & Han, Y. (2019). Exploring Goodreads reviews for book impact assessment. *Journal of Informetrics*, *13*(3), 874–886. https://doi.org/10.1016/j.joi.2019.07.003

[10] Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, *28*, 45–59. https://doi.org/10.1016/j.inffus.2015.08.005

[11] Elghannam, F. (2019). Text representation and classification based on bi-gram alphabet. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2019.01.005

[12] Chalothorn, T., & Ellman, J. (2015). Simple approaches of sentiment analysis via ensemble learning. In *Lecture Notes in Electrical Engineering* (Vol. 339, pp. 631–639). https://doi.org/10.1007/978-3-662-46578-3_74

[13] Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. *IEEE Access*, *8*, 23522–23530. https://doi.org/10.1109/ACCESS.2020.2969854

[14] Zeng, D., Dai, Y., Li, F., Wang, J., & Sangaiah, A. K. (2019). Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. *Journal of Intelligent & Fuzzy Systems*, *36*(5), 3971–3980. https://doi.org/10.3233/JIFS-169958

[15] Khan, K., Baharudin, B., Khan, A., & Ullah, A. (2014). Mining opinion components from unstructured reviews: A review. *Journal of King Saud University - Computer and Information Sciences*, *26*(3), 258–275. https://doi.org/10.1016/j.jksuci.2014.03.009

[16] Hussein, D. M. E.-D. M. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, *30*(4), 330–338. https://doi.org/10.1016/j.jksues.2016.04.002

[17] Moraes, R., Valiati, J. F., & Gavião Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621–633. https://doi.org/10.1016/j.eswa.2012.07.059

[18] McAuley, J., & Leskovec, J. (2013). From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. *WWW 2013 - Proceedings of the 22nd International Conference on World Wide Web*, 897–907. https://doi.org/10.1145/2488388.2488466

[19] Willett, P. (2006). The Porter stemming algorithm: then and now. *Program*, *40*(3), 219–223. https://doi.org/10.1108/00330330610681295

[20] Xie, F., Wu, X., & Zhu, X. (2017). Efficient sequential pattern mining with wildcards for keyphrase extraction. *Knowledge-Based Systems*, *115*, 27–39. https://doi.org/10.1016/j.knosys.2016.10.011

[21] Gencosman, B. C., Ozmutlu, H. C., & Ozmutlu, S. (2014). Character n-gram application for automatic new topic identification. *Information Processing & Management*, *50*(6), 821–856. https://doi.org/10.1016/j.ipm.2014.06.005

[22] Schmidt, C. W. (2019). Improving a tf-idf weighted document vector embedding.

[23] Ren, J. (2012). ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Knowledge-Based Systems*, *26*, 144–153. https://doi.org/10.1016/j.knosys.2011.07.016

[24] Appel, O., Chiclana, F., Carter, J., & Fujita, H. (2016). A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Systems*, *108*, 110–124. https://doi.org/10.1016/j.knosys.2016.05.040

[25] 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. https://doi.org/10.1016/j.eswa.2016.03.028