Improve Term Weighting for Text Classification

Abdulmohsen Algarni
King Khalid University, College of Computer Science, KSA

Abstract:
A large number of features can be extracted from text documents. The extracted features are mixed between positive, negative, and noise features. Improving the quality of extracted features can be a challenging task. The feature selection technique used in text classification is the team-based approach. The term "Frequency-Inverse Document Frequency" (TF-IDF) is widely used to extract all features from text documents. Based on the weight of the extracted features, the most important features can be selected. However, the selected features are based on the frequency of the term in the documents regardless of the importance of the feature. In this paper, we proposed a new method based on TF-IDF to improve the quality of extracted features and revise the weight. The extracted terms from the text are classified based on their importance. Then, the weight of the features can be revised based on the class that the feature belongs to. The proposed model shows significant improvement in the evaluation measures with an average of 3.6 in F-measure.

Keywords: Feature Selection, Classification, term weighting
DOI: 10.24297/j.cims.2023.11.5

1. Introduction

In the era of big data, huge amounts of data are available in electronic format in different domains, such as health care, education, online shopping, and security. The data are known as high-dimensional data, especially text data [15]. Then, extracting high-quality knowledge from high-dimensional data is a challenge. Applying data mining and machine-learning methods to high-dimensional data will lead to that data becoming sparser in high-dimensional space [13], [15].

Text classification is used to assign text to different classes based on the predefined set of categories [10,25,26]. To classify a set of documents D, the classifier T needs to be built first. The classifier T needs to train by using a labelled set of documents D [6]. The effectiveness of the classifier is affected by many factors. One of them is the quality of training documents and the quality of features selected and extracted [2]. However, a massive number of features can be extracted from text. Some of those features are meaningless or misleading to the classifier. The simple process of text classification can be illustrated in Figure 1. Therefore, improving the feature selection process can play a significant role in the effectiveness of the classifier. Feature selection helps to directly select a subset of relevant features for model construction [12,16].

One of the main tasks that can affect the text classification model is the pre-processing process and feature selection and extraction methods [23]. Selecting the right features (terms)to represent the documents is a very challenging task. The extracted features represent the documents in vector space. In vector space, each term has a weight, and the weight of the term determines the importance of the term. As a result, the weighting methods can play a significant rule in the effectiveness of the classifiers. Thus, a challenge in automatic text classification is the effective weighting and representation of text [23].

Feature extraction methods extract a huge number of features. Some of the extracted features are useless because they are misleading the classifiers in categorizing the documents into different classes due to their weak discriminatory power [2]. Generally, those features are shared between all the text in different categories. Sometimes, their term frequency is the same in all classes.

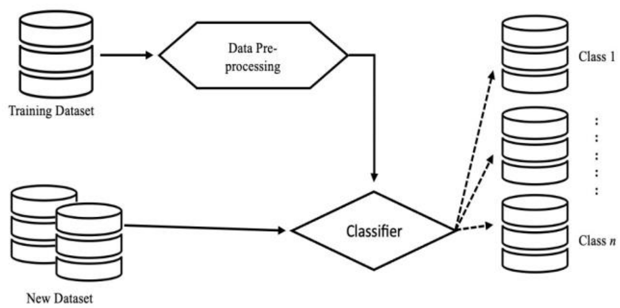


Figure 1: Documents Classification process.

Different weighting methods have been developed over the years. Most of them used the same weighting technique for all the features without any discriminative of the important features. The extracted features from a text can be categorized into three categories: features that appear only in positive documents, features that appear only in negative documents, and features that appear in both positive and negative documents. Most of the features appear in both positive and negative classes. Those features normally appear equally in both classes which makes them useless. Then, not considering these features in the weighting method can affect the classifier's performance.

In this paper, we introduce a technique that revises the selected features from the text and groups them according to their importance. Based on the group of the terms, their weight can be revised.

The remainder of this paper is organized as follows: sections 2 and 2 present the literature review and related work. The proposed scheme is discussed in section 4. Section 5 shows the experimental settings, the result, and the discussion. Finally, section 6 contains the conclusion and directions for future work.

2. RELATED WORK

Text classification has been used wildly in real-life applications such as spam email filtering and webpage classification. The classifier learns to classify a new text based on pre-specified label data [27] Generally, text contains a large number of unique terms. Using all the terms to train the classifiers can give a poor result [17,22]. Feature selection techniques can be used to select the most important terms that can be used to train the system.

Feature selection and feature extraction techniques can improve the effectiveness of the learning process and the efficiency of the classifiers. They help to reduce the dimensions of the data and focus on the most important features. Even in low-diminution data, both feature selection and feature extraction can still play significant roles in improving the effectiveness of the model and avoiding over-fitting problems.

Generally, the text contains a large number of features, most of which can be categorized as noise or redundant features [11]. Therefore, building a text classification model is a sophisticated process. It contains some critical steps that can affect the performance of the classifier, such as data pre-processing, transformation, feature selection, and feature extraction [18]. Feature selection and extracting is known as the first steps to build a classifier after the pre-processing steps. Term Frequency-Inverse Document Frequency (TF-IDF) is one of the popular methods in feature extractions [14].

Many text representation models have been developed over the years. The simplest and the most popular method is the bag of words. The method used to select those words is the Term Frequency (TF) method. More advanced methods can be used to give more accurate results such as the term Frequency-Inverse Document Frequency (TF-IDF). Whereas, inverse Document Frequency (IDF) is to reduce the effect of general words in a set of documents [14].

However, TF-IDF can extract a large number of features, most of which are considered noise features. Some studies contend that most of the dataset contains about 40% noise [11]. Therefore using all the extracted features as they are to train the classifiers can affect the performance of the classifier.

In text classification, different numbers of algorithms have been developed [14]. The text classification algorithms can be grouped into two: traditional machine learning and deep learning [20]. Each category has its own advantages and disadvantages. The advantage of traditional models is that they require less training datasets and less computational power compared to the deep learning approach.

2.1 Document Representation and Term Weighting

Both features selection and features extraction are critical steps to improve the effectiveness of the ML algorithm [28]. To deal with text documents, the terms need to be changed to a specific format that the model can deal with. The most common format is the vector space model (VSM). Whereas each document transfers to a vector that combines the terms and the weight of each term, such as d ={t1, t2,t3, ..., tn}, where d is a document that contains the n terms, and wi is the weight value of the ith term. The weight of each term represents the importance of the terms in VSM. The weighting scheme used to calculate the weight of the terms can play an important role in the effectiveness of the classifier. The weighting scheme can be grouped into two groups: Local Factor and Global Local Factor.

On one hand, the local factor focuses on the contribution of the term in each individual document [9]. Term frequency (tf) is known as the most used local factor method to calculate the weight of each term in a different document. It calculates how many times that specific term occurs in the document [24].

On the other hand, the global factor focuses on the weight of the terms in the whole collection [9]. However, the global factor method is affected by the local factor methods [8]. The global factor methods can be grouped into two groups: unsupervised term weighting methods (UTW) and supervised term weighting methods (STW).

UTW Schemes: In unsupervised term weighting methods the category of the documents is ignored during the calculation of the weight of term ti in documents. In other words, the weighting of the term ti does not consider the classes of that term belong to. Therefore, the weight considers only the frequency of the occurrences in all the training datasets. It used to balance the term frequency using inverse document frequency as follows:

$$tf - idf = TF(t_i, d_j) \times log\left(\frac{N}{df(t_i)}\right) \quad (1)$$

where TF is the frequency of term ti in document dj, N is the number of training documents, and df is the number of documents d that contain term ti in the dataset. TF-IDF has been used widely as a feature extraction in many text classification methods [3]–[5], [19,21]

STW Schemes: The STW considers the classes of the documents while calculating the weight. Therefore, this process is known as a supervised task. It used different term weighting methods to control the term weighting process [9]. Different STW methods are used to overcome the limitations of UTW. It is proposed to consider the classes of the extracted features.

The Proposed STW Scheme

The UTM limitation is that the weight of the terms is calculated based on the frequency of the term in the entire dataset regardless of the category of the documents. Therefore, using the STW method would help to overcome this problem. TF-IDF method used to calculate the weight of specific term ti in set of documents D as shown in Eq.1.

The most popular term weighting scheme is TF-IDF, which assesses the importance of a term ti in a document based on the entire corpus, as show in Eq.1.

The extracted terms using *TF-IDF* can be group into 3 classes:
General terms group *G*, the positive specific terms group *T +*, and the negative specific terms group *T-*. The classification rules show can be generalized as follow:

The extracted terms using *TF − IDF* can be grouped into 3 classes: General terms group G, the positive specific terms group *T+*, and the negative specific terms group *T−*. The classification rules show can be generalized as follows:

Common = {t ∈ T | t ∈ D+ & t ∉ D−},
T+ = {t ∈ T | t ∈ D+ & t ∉ D−}, and
T− = {t ∈ T | t ∈ D− & t G ∉ D+}.

It is easy to verify that *G ∩ T+ ∩ T− = ∅*. Therefore, *{Common, T+, T−}* is a partition of all terms.

This research proposed a novel STW model called term frequency-term discrimination ability *(TF-TDA)*. The proposed model aims to improve the quality of features extracted from text by grouping the terms into more than three groups. Then, the weight of the terms can be revised in each group.

3.1 Term Classification and Wight revising

We believe that the terms that appear in Common classes can be categorized into three classes. The grouping function can consider the frequency of the terms in a positive document D+ and the term frequency in D-. Whereas the term t appears more to the positive group can be more important to the positive classes. On the other hand, the term t that appears more in negative classes can be more important to the negative Classes. As a result, each term ti has two weights. The first weight is the weight of the term in positive class PE(ti). The second weight is the weight of the term ti in negative class NE(ti). Both weights can be calculated used the following equations:

$$P_E(t_i) = \frac{a(t_i)}{N^+} \times 100 \quad (3)$$

$$N_E(t_i) = \frac{c(t_i)}{N^-} \times 100 \quad (4)$$

Whereas a is the frequency of term ti in positive classes, and *N+* is the number of positive training documents, c is the frequency of term ti in negative classes, *N−* is the number of negative training documents. In case of term ti appears positive classes only then the value of *NE(ti) = 0*. If the term *ti* appears in negative classes only then the value of *PE(ti) = 0.*

Based on the weight of the terms in each class, the distance of term *ti* to each class can be calculated using the following equation:

$Var(t_i) = P_E(t_i) - N_E(t_i)$ (5)

Consequently, the common terms can then be classified into *Freq+ or Freq−*, or *G* by satisfying

$$Common(t) = \begin{cases} t \in Freq^{+}, & \text{if } Var \geq k \\ t \in Freq^{-}, & \text{if } Var \leq -k \\ t \in G, & \text{if } |Var| \leq k \end{cases} \quad (6)$$

the following constraints:

where k is the threshold parameter

This research aims to develop a novel STW scheme called Term Frequency-Term Discrimination Ability (TF-TDA). The proposed group the terms in more than three categories *(T+, T−, and Com).*

Based on the grouping classes, the term' s weight can be revised based on the distinguishing ability, especially the imbalanced

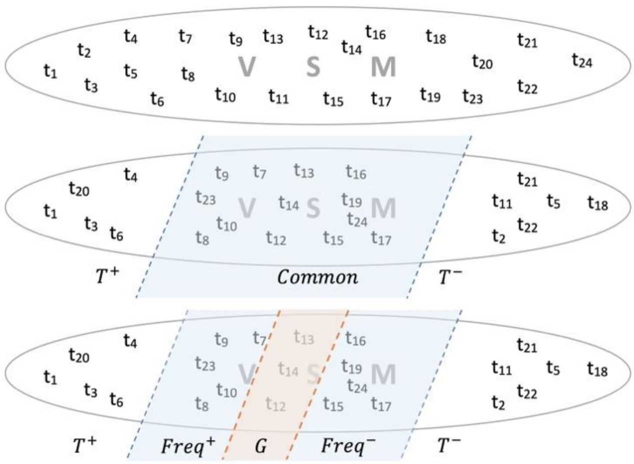


Figure 2: The classification of terms in the proposed model.

dataset. The distinguishing ability of the term ti can be calculated using class priority *(clsPrior*), formulated as follows:

$$clsPrior(c_k, t_i) = \frac{N^{c_k}}{N} \times \frac{|C|}{cf(t_i)} \quad (7)$$

Where N is the number of training documents in the dataset, |*C*| is the number of binary classes in the dataset, and |C| = 2, *cf(ti)* is the number of classes that *ti* appears in.

In this paper, we apply a binary classification to the data set. Therefore, the revised weight for each group of terms can be formalized as follows:

IF ($t_i \in T^+$):

$$wight(t_i) = TF(t_i, d_j) \times log_2\left(2 + \frac{P_E(t_i)}{max(1, N_E(t_i))} \times \left(\frac{N^+}{N} \times 2\right)\right)$$

IF ($t_i \in T^-$):

$$wight(t_i) = TF(t_i, d_j) \times log_2\left(2 + \frac{N_E(t_i)}{max(1, P_E(t_i))} \times \left(\frac{N^-}{N} \times 2\right)\right)$$

IF ($t_i \in Freq^+$):

$$wight(t_i) = TF(t_i, d_j) \times log_2\left(2 + \frac{P_E(t_i)}{max(1, N_E(t_i))} \times \left(\frac{N^+}{N} \times 1\right)\right)$$

IF ($t_i \in Freq^-$):

$$wight(t_i) = TF(t_i, d_j) \times log_2\left(2 + \frac{N_E(t_i)}{max(1, P_E(t_i))} \times \left(\frac{N^-}{N} \times 1\right)\right)$$

IF ($t_i \in G$):

Table 1: TF-TDA Algorithem.

TF-TDA(D)

Input: A training set, $D = D^+ \cup D^-$,
Output: extracted features $< T >$,
Empirical parameter k;

Method:
$n = |D^+|, m = |D^-|$;
$T =$ *TF-IDF*(D);
$T^- = \{t|t \in T, t \in D^-, t \notin D^-\}$;
$T^+ = \{t|t \in T, t \in (D^+ \cap D^-)\}$
$Common = \{t|t \in T, t \in D^-, t \notin D^-\}$;
foreach $t \in Common$ **do**
  $P_E(t_i) = \frac{a(t_i)}{N^+} \times 100$
  $N_E(t_i) = \frac{c(t_i)}{N^-} \times 100$
  $Var(t_i) = P_E(t_i) - N_E(t_i)$
    if ($Var(t_i) \geq k$)
    $Freq^+ = Freq^+ \cap t_i$
    if ($Var(t_i) \leq -k$)
    $Freq^- = Freq^- \cap t_i$
    if ($|Var(t_i)| \leq k$)
    $G = G \cap t_i$
foreach $t \in T^+$ **do**
  weight(t)= TF(t_i,d_j) $\times log_2\left(2 + \frac{P_E(t_i)}{max(1, N_E(t_i))} \times \left(\frac{N^+}{N} \times 2\right)\right)$
foreach $t \in T^-$ **do**
  weight(t)= TF(t_i,d_j) $\times log_2\left(2 + \frac{N_E(t_i)}{max(1, P_E(t_i))} \times \left(\frac{N^-}{N} \times 2\right)\right)$
foreach $t \in Freq^+$ **do**
  weight(t)= TF(t_i,d_j) $\times log_2\left(2 + \frac{N_E(t_i)}{max(1, P_E(t_i))} \times \left(\frac{N^-}{N} \times 2\right)\right)$
foreach $t \in Freq^-$ **do**
  weight(t)= TF(t_i,d_j)$\times log_2\left(2 + \frac{P_E(t_i)}{max(1, N_E(t_i))} \times \left(\frac{N^+}{N} \times 1\right)\right)$
$T = T^+ \cup T^- \cup Freq^+ \cup Freq^- \cup G$;

$wight(t_i) = TF(t_i, d_j)$

The proposed algorithm presented in Table 1. from step 1 to step 5 is the extraction of features from text using TF-IDF, and the extracted features grouped into three groups *(T+,Common, T−)*. Then from step 6 to step 15, shows the process of classifying the terms in *Common* group into three classes (*Freq+,G,Freq−*). Then the weight was revised for each term in the *Freq+,G* and *Freq −*class from step 16 to 23.

The time complexity of TF-TDA is mainly decided by the steps of classifying the common terms into 3 groups (*Freq+,G,Freq−*). Then, the revision process of the weight of 4 out of the 5 groups (*T+,Freq+,G,Freq−,T−*) was undertaken. Therefore, the time complexity of TF-TDA can be calculated as O($mlog^m$), where m = $|T|$.

3. EVALUATION

The proposed model, TF–IDF, aims to classify the extracted features into 5 classes. Then the features weight can be revised based on the classes of each term. The aim of that is to improve the effectiveness of the classifier. Therefore, the proposed model, *TF–IDA*, is compared with the traditional *TF–IDF*.

TABLE II: F-measure result for the models used.

Model	Dataset	TF-IDF	TF-TDA	TF-IDF*
M-NB	MARSA (Social)	82.31%	85.52%	3.21%
SVM	MARSA (Social)	79.23%	83.22%	3.99%
* The improvement percentage obtained by TF-TDA compared to other schemes				

4.1 Datasets

In order to evaluate the proposed model, we use the Multidomain Arabic Resources for Sentiment Analysis (MARSA) dataset [1]. From all the domains, we choose the social domain. The social dataset concentrated on issues affecting Saudi society. Therefore, hashtags were created about social issues, such as royal orders, the Saudi budget, issues affecting the income of Saudi citizens, and others. They contain about 7523 tweets in the following categories: 2499 tweets were classified as positive, and 5024 tweets were negative. In order to use the dataset, the pre-processing steps have been applied for each tweet. That includes stopwords removal and stemming. The dataset is divided into two groups: training and testing. The number of documents in each group is presented in Table I and the percentages are shown in Figure 3.

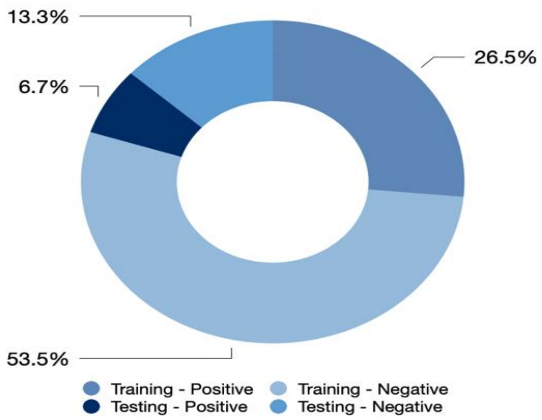


Figure 3: The distribution of dataset in this

4.2 Evaluation Measures

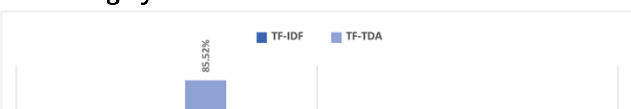
In any data mining process, evaluation is a vital point. Commonly used evaluation measures are precision, recall, and F-measure. Precision basically shows, out of the instances classified as positive by the model, how many twitters are actually positive. Recall, however, shows how many tweets the model classified as positive out of the instances that are actually positive. In this context, evaluating both measures and then selecting the one with the higher value is not appropriate in some cases. Thus, there is another technique called F-measure, also known as F-measure, which is a metric that takes both precision and recall into account. The F-measure gives equal weight to precision and recall. This means that if any recall or precision is low, then the F-measure will tend to be low as well. F-measure is not the proper metric to evaluate models in cases where the models demand recall or precision to be higher than another. Precision, recall, and F-measure can be calculated using the following equations:

$$precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - measure = \frac{2 \times precesion \times recall}{precision + recall}$$

Where *TP* (true positive) is the case in which a student is predicted as at-risk and the actual class of the student was also at risk, *FP* (false positive) is the case in which the student is predicted as



not at-risk and the actual class was at risk, *TN* (true negative) is the case in which the student is predicted as not at-risk and the actual class was not at risk, and *FN* (false negative) is the case in which the student is predicted as at risk and the actual class was not at risk.

Figure 4: Comparing the pressed model TF-TDA with traditional TF-IDF in SVM and NB.

TABLE III: Number of unique features (including common and pure features) for each dataset.

Dataset	|Com|	|T+|	|T−|	|AllFeatures|
MARSA (Social)	2249	3211	862	6322

4.3 Result

A vector space model (VSM) was used for term representation, using words as features. The main method that is used to extract terms is the traditional method TF. Table III presents the number of extracted terms from the dataset. Moreover, the number of terms that appear in different classes is shown in Table III. Based on TF, the proposed model TF − TDA is applied in two steps:

- Grouping the extracted terms into 5 different groups.
- Revised the weight of the extracted terms.

The proposed weighting scheme will be compared with the traditional TF-IDF model using two classifiers. Both support vector machine (SVM) and naive bayes (NB) have been used to

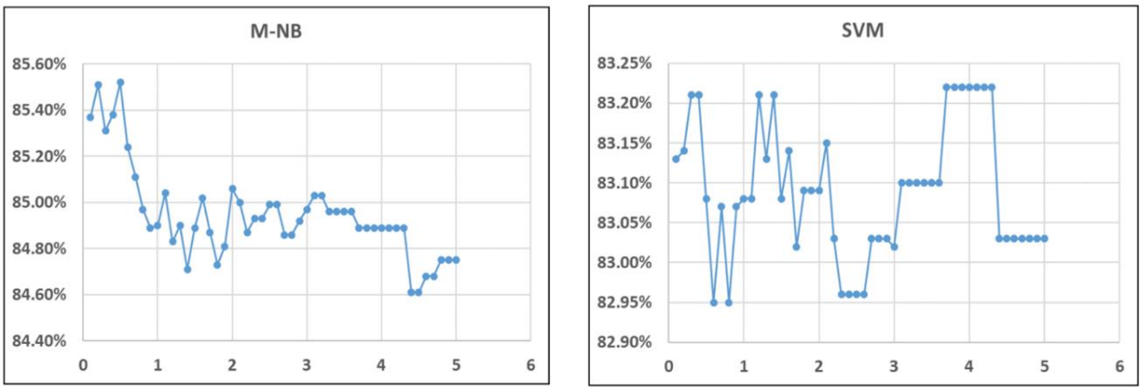
TABLE IV: Statistical Tests Results

Dataset	Model	TF-TDA vs.TF-IDF
		P-value
MARSA	M-NB	0.00052
(Social)	SVM	0.000233

The proposed weighting scheme will be compared with the traditional TF-IDF model using two classifiers. Both Support Vector Machine (SVM) and Naive Bayes(NB) have been used to evaluate the model. The classifiers were tested using TF-IDF and TF-TDA. As shown in Table II, the proposed model gave better results in the F-measure. The percentage of improvement in both SVM and NB was 3.6%=(3.21+3.99)/2 on average. The result is presented in Figure 4. Based on the result, the proposed model gave a better result, and the NB algorithm using TF_TDA gave the best results.

We use statistical tests to calculate the *P-value* to measure the significance of the improvement. This value is known as the probability of no difference between two set of obtaining results (null hypothesis). The *P-value* can indicate if the null hypothesis can be rejected or not. In this paper, the *P-value* is calculated using McNemar' s statistical test [7], where the significance level is 0.05. In Table IV, the *P-value* result shows that it is less than 0.05. Therefore, the null hypothesis can be rejected. In other words, the improvement of the result using TF-IDF is significant, whereas P – value = 0.00052 < 0.05 in the maximum value.

As shown in Eq. IV-A, the value of empirical parameters, like k, that are used to determine the boundary of each class (*Freq+, Freq−, G*) can affect the result. The value of k can affect the number of terms in each class. Table V shows the distribution of the terms in all classes using the optimal value of k. Figure 5 shows the F-measure result for different value of *k*.



(a) F-measure of different *k* values using NB model. (b) F-measure for different *k* values using SVM model

Figure 5: Results of All considered *k* values in different models.

TABLE V: Optimal k in different.

Model	k	|G|	|Freq+|	|Freq−|	|T+|	|T−|	F1-score
M-NB	0.5	1995	111	143			85.52%
SVM	3.7	2235	12	2	3211	862	83.22%

4. CONCLUSION

Different methods can be used to extract terms and present features in the vector space model. Most existing techniques can be categorized into one of the two groups: local factor and global factor. In this paper, we proposed a novel TF-TDA scheme to improve the effectiveness of text classification methods. The proposed model works in two stages as follows:

- Term classification into different classes based on the importance of each class.
- Revise the weight of the extracted terms in each class according to the importance of each class.

The result of the proposed model shows a significant improvement in F-measures in both SVM and NB models. The percentage of improvement is about 3.6% on average. Using statistical methods to show the significance of the results gives p-value = 0.00052 using NB and p-value = 0.000233 using SVM. More research can be conducted in multiple classes. Moreover, Applying the same approach in different domains, such as images, may improve the result.

5. ACKNOWLEDGEMENTS

The author would like to thank the Deanship of Scientific Research at Shaqra University for supporting this research

REFERENCES

1. Alowisheq, n. Al-twairesh, m. Altuwaijri, a. Almoammar, a. Alsuwailem, t. Albuhairi, w. Alahaideb, and s. Alhumoud. Marsa: multidomain arabic resources for sentiment analysis. Ieee access, 9:142718– 142728, 2021.
2. Alshehri and a. Algarni. Tf-tda: a novel supervised term weighting scheme for sentiment analysis. Electronics, 12(7), 2023.
3. M. Altawaier, s. Tiun, et al. Comparison of machine learning approaches on arabic twitter sentiment analysis. International journal on advanced science, engineering and information technology, 6(6):1067–1073, 2016.
4. N. Aslam, k. Xia, f. Rustam, a. Hameed, and i. Ashraf. Using aspectlevel sentiments for calling app recommendation with hybrid deeplearning models. Applied sciences, 12(17):8522, 2022.

5. N. Aslam, k. Xia, f. Rustam, e. Lee, and i. Ashraf. Self voting classification model for online meeting app review sentiment analysis and topic modeling. Peerj computer science, 8:e1141, 2022.
6. X. Deng, y. Li, j. Weng, and j. Zhang. Feature selection for text classification: a review. Multimedia tools and applications, 78:3797– 3816, 2019.
7. T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation, 10(7):1895– 1923, 1998.
8. T. Dogan and a. K. Uysal. On term frequency factor in supervised term weighting schemes for text classification. , 44(11):9545–9560, 2019.
9. G. Domeniconi, g. Moro, r. Pasolini, and c. Sartori. A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf. Idf. In international conference on data management technologies and applications, pages 39–58. Springer, 2015.
10. S. Dumais, j. Platt, d. Heckerman, and m. Sahami. Inductive learning algorithms and representations for text categorization. In proceedings of the seventh international conference on information and knowledge management, pages 148–155, 1998.
11. U. M. Fayyad, g. Piatetsky-shapiro, and r. Uthurusamy. Summary from the kdd-03 panel: data mining: the next 10 years. Acm sigkdd explorations newsletter, 5(2):191–196, 2003.
12. Guyon and a. Elisseeff. An introduction to variable and feature selection. Journal of machine learning research, 3(mar):1157–1182, 2003.
13. T. Hastie, r. Tibshirani, j. H. Friedman, and j. H. Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.
14. K. Kowsari, k. Jafari meimandi, m. Heidarysafa, s. Mendu, l. Barnes, and d. Brown. Text classification algorithms: a survey. Information, 10(4), 2019.
15. J. Li, k. Cheng, s. Wang, f. Morstatter, r. P. Trevino, j. Tang, and
16. Liu. Feature selection: a data perspective. Acm comput. Surv., 50(6), dec 2017.
17. H. Liu and h. Motoda. Computational methods of feature selection. Crc press, 2007.

18. H. Liu and l. Yu. Toward integrating feature selection algorithms for classification and clustering. Ieee transactions on knowledge and data engineering, 17(4):491–502, 2005.

19. M. M. Mironczuk and j. Protasiewicz. A recent overview of the state-of-´ the-art elements of text classification. Expert systems with applications, 106:36–54, 2018.

20. M. Mujahid, e. Lee, f. Rustam, p. B. Washington, s. Ullah, a. A. Reshi, and i. Ashraf. Sentiment analysis and topic modeling on tweets about online education during covid-19. Applied sciences, 11(18):8438, 2021.

21. J. T. Pintas, l. A. Fernandes, and a. C. B. Garcia. Feature selection methods for text classification: a systematic literature review. Artificial intelligence review, 54(8):6149–6200, 2021.

22. F. Rustam, i. Ashraf, a. Mehmood, s. Ullah, and g. S. Choi. Tweets classification on the base of sentiments for us airline companies. Entropy, 21(11):1078, 2019.

23. F. Sebastiani. Machine learning in automated text categorization. Acm computing surveys (csur), 34(1):1–47, 2002.

24. N. Shanavas. Graph-theoretic approaches to text classification. Phd thesis, ulster university, 2020.

25. H. Wu, x. Gu, and y. Gu. Balancing between over-weighting and under-weighting in supervised term weighting. Information processing & management, 53(2):547–557, 2017.

26. Y. Yang. An evaluation of statistical approaches to text categorization. Information retrieval, 1(1-2):69–90, 1999.

27. Y. Yang and x. Liu. A re-examination of text categorization methods. In proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval, pages 42–49, 1999.

28. S. Zhao, h. Yao, s. Zhao, x. Jiang, and x. Jiang. Multi-modal microblog classification via multi-task learning. Multimedia tools and applications, 75:8921–8938, 2016.

29. Zheng and a. Casari. Feature engineering for machine learning: principles and techniques for data scientists. " o' reilly media, inc.", 2018.