International Journal of Innovative
Computing, Information and Control
Volume **16**, Number **5**, October **2020**

ICIC International ©2020 ISSN 1349-4198
pp. **1593–1609**

# A LABEL-ORIENTED APPROACH FOR TEXT CLASSIFICATION

Manh Hung Nguyen

Faculty of Computer Sciences
Posts and Telecommunications Institute of Technology (PTIT)
No. 122, Hoang Quoc Viet Street, Cau Giay District, Hanoi 084, Vietnam
mhnguyen@ptit.edu.vn; nmh.nguyenmanhhung@gmail.com

Received May 2020; revised August 2020

Abstract. *Text classification is a well known problem in the machine learning community. A widely used approach is that based on the Term Frequency – Inverse Document Frequency (TF-IDF) feature. This feature represents very well the characteristic of a text. However, this feature could not clearly represent the relationship from a text to its assigned label. This paper presents a Label-Oriented (LO) approach for text classification problem. This approach takes account of the relationship between a text and its assigned label by introducing a new feature label-oriented score. This score represents the level of the importance of the term regarding all terms and texts assigned to the label compared to all terms and texts unassigned to the label. In the training phase, this model calculates the label-oriented score of each term to a label. In the testing phase, the sum of this score of all terms in a text will help us to determine whether the text should be assigned to the label or not. The proposed model is then evaluated in two cases: short and regular texts. The experiment results indicate that the proposed model is significantly better than baseline models on the considered datasets.*

**Keywords:** Machine learning, Supervised machine learning, Label-oriented score, Text classification

1. **Introduction.** The problem of text classification is already popular in the machine learning community. In it, a set of texts which are already classified with a label class, called *training set*, will be used to extract some common features of texts of the same label. If there is a new text $t$, the assignment of a label to the text $t$ is based on the relationship among the text itself and the texts in the training set.

A widely used approach is that based on the *TF-IDF feature vector* of texts [37]. This feature counts the Term Frequency (TF) in a text and the Inverse Document Frequency (IDF) of a term regarding all texts in the training set. Therefore, the TF-IDF value of a term depends only on the text (by the TF portion) and the training set (by the IDF portion). In other words, the TF-IDF value of a term does not depend on the label of the text. Let us consider an example. There is a text $t$ in a set $T$ of text, and its label $l_1$. In the training phase, the *TF-IDF feature vector* of $t$ is calculated based only on the appearance of terms in $t$ and the texts in $T$ which contain the considered term. This vector is calculated without any consideration of its label $l_1$. Consequently, if the label is changed to $l_2 \neq l_1$, the *TF-IDF feature vector* of $t$ is still the same as that in case of label $l_1$. Intuitively, in the text classification problem, when a text is assigned to a label, its label is more or less meaningful to the text. When the label of a text changed, the *feature* of the text is thus more or less changed. Therefore, it may be better if the *feature vector* of a text counts also the relationship between the text and its assigned label.

DOI: 10.24507/ijicic.16.05.1593

This paper aims to take account of this relationship between the text and its assigned label by proposing a new feature called *Label-Oriented (LO) score*. This score represents the level of the importance of the term regarding all terms and texts assigned to the label compared to all terms and texts unassigned to the label. The *label-oriented score* of a term to a label is calculated as follows: the training set is divided into two subsets: a subset of texts which are assigned to the label, and the other subset of texts which are unassigned to the label. Then, the frequency of text – which contains the term – in the subset assigned to the label will be compared to that of text containing the term in the subset unassigned to the label. The bigger this score (more positive), the higher the possibility that a text containing this term belongs to the subset of texts which are *assigned* to the label. And vice versa, the smaller this value (more negative), the higher the possibility that a text containing this term belongs to the subset of texts which are *unassigned* to the label. In the training phase, the *label-oriented score* is calculated for all terms which appear in texts in the training set. The training result is a vector of *label-oriented score* for all terms in training set for each label. In the testing phase, the sum of this *label-oriented score* of all terms in a text (regarding a given label) will help us to determine whether the text should be assigned to the label or not. The proposed model is then evaluated in two cases: (i) short text with the EmoLex dataset [30], and (ii) regular texts with the 20 Newsgroups dataset [23].

The paper is organized as follows. Section 2 presents the related works to this paper. Section 3 presents the model based on the label-oriented score. Section 4 presents the evaluation of the proposed model. Section 5 is the conclusion.

## 2. **Related Works.**

There are many models proposed to solve this problem. Let us consider these models on two aspects (Table 1): (i) the considered technical features; (ii) the classifier used.

Regarding the technical feature of models, there are two main tendencies: semantic-based approaches and statistic-based approaches. Firstly, the semantic-based approaches try to extract the meaning of words, sentences in texts to measure the similarity among them. For instance, an ontology is used to classify texts [47]; or a combination of both lexical and semantic features to classify texts [41]; or a combination of the semantic and statistic approaches [3, 15, 26]. Secondly, the statistic-based approaches try to extract the statistical features of texts to estimate the distance among texts. A classical and widely used technique in this tendency is that based on the TF-IDF feature. In it, each text or document is split into a set of terms. Then the TF-IDF value of each term is calculated. The set of TF-IDF value corresponding to the set of original terms of a text in the *training set* will form the *feature vector* of the text. The *feature vector* of the new text is also determined in the same manner. The classification of the new text is then based on the *distance* between the new text to each text in the *training set*. This *distance* is considered as the distance between the *feature vector* of the two concerned texts, i.e., Danesh et al. [10], Erkan et al. [13], Nigam et al. [32], Rossi et al. [36], Joulin et al. [20], Al-Anzi et al. [2].

Regarding the classifier used in models, there are also two main tendencies: using classical classifier-based or define their own classifier. In the first tendency, any of (or extended of any) classical classifiers such as Naive Bayes (NB) [19], Support Vector Machine (SVM) [7], k-Nearest Neighbors (kNN or IBk) [1], C4.5 [34] could be used. For instance, NB is used in model of Rossi et al. [36], Vilar et al. [38], Wei et al. [40], Zhang et al. [44]. SVM is used in model of Bekkerman et al. [4], Liu et al. [28], Wang and Chiang [39]. K-Means is used in the work of Hendry and Chen [17]. A neural network is also used in the work

TABLE 1. Summary of related works

| Model | Technique | | Classifier | |
|---|---|---|---|---|
| | Semantic | Statistic | Classical | Newly proposed |
| Albitar et al. [3] | ✓ | | | |
| Bekkerman et al. [4] | | | ✓ | |
| Bouras et al. [6] | | | | ✓ |
| Chen et al. [8] | | | | ✓ |
| Dai et al. [9] | | | ✓ | |
| Danesh et al. [10] | | ✓ | ✓ | |
| De Souza et al. [11] | | | ✓ | |
| Devlin et al. [12] | | ✓ | | ✓ |
| Erkan et al. [13] | | ✓ | ✓ | |
| Glinka and Zakrzewska [14] | | | | ✓ |
| Gliozzo et al. [15] | ✓ | | | ✓ |
| Goldberg et al. [16] | | | | ✓ |
| Joulin et al. [20] | | ✓ | ✓ | |
| Kibriya et al. [21] | | | ✓ | |
| Kim [22] | | | | ✓ |
| Lau et al. [24] | | | | ✓ |
| Li et al. [25] | | | | ✓ |
| Liu et al. [28] | | | ✓ | |
| Mikolov et al. [29] | | | | ✓ |
| Nam et al. [31] | | | ✓ | |
| Nigam et al. [32] | | ✓ | ✓ | |
| Peters et al. [33] | | ✓ | | ✓ |
| Read et al. [35] | | | | ✓ |
| Rossi et al. [36] | | ✓ | ✓ | |
| Vilar et al. [38] | | | ✓ | |
| Wang and Chiang [39] | | | ✓ | |
| Wei et al. [40] | | | ✓ | |
| Yang et al. [41] | ✓ | | | |
| Yu et al. [42] | | | | ✓ |
| Yu et al. [43] | | | | ✓ |
| Zhang et al. [44] | | | ✓ | |
| Zhang and Zhou [45] | | | ✓ | |
| Zhang et al. [46] | | | | ✓ |
| Zhou and El-Gohary [47] | ✓ | | | |

of De Souza et al. [11], Nam et al. [31], Zhang and Zhou [45], Hendry et al. [18], Li et al. [27].

Moreover, some authors could improve some classical classifier for their model. For instance, Danesh et al. [10] proposed three improvements using Decision Template, or Voting, or Ordered Weighted Averaging (OWA); Erkan et al. [13] proposed a model with a Harmonic function; or the Expectation Maximization (EM) of Nigam et al. [32]; the Multinomial Naive Bayes of Kibriya et al. [21]. Dai et al. [9] proposed a transfer-learning algorithm for text classification based on an EM-based NB.

In the second tendency, some authors could propose their own classifier. For instance, Lau et al. [24] used best topic word selection and re-ranking model. Bouras et al. [6] combined the co-operation of the categorization and summarization mechanisms. Goldberg et al. [16] used crowd-sourcing techniques. Gliozzo et al. [15] proposed a text-categorization bootstrapping algorithm. Yu et al. [43] used the variable precision neighborhood rough sets. Read et al. [35] used binary relevance-based methods. Li et al. [25] proposed a joint learning algorithm. Yu et al. [42] developed techniques that exploit the structure of specific loss functions, such as the squared loss function. Glinka and Zakrzewska [14] proposed Labels Chain (LC) algorithm based on relationship between labels. An approach of text-to-vector is proposed by Mikolov et al. [29]. Joulin et al. [20] proposed their FastText model. Conventional and recurrent neural networks are used in the model of Kim [22], Chen et al. [8], Zhang et al. [46]. In the same approach, Peters et al. [33] proposed ELMo model and Devlin et al. [12] proposed BERT model based on the convolution and recurrent neural networks.

However, one of the main limits of these models is that they use only the *TF-IDF feature vector* of texts. This vector is independent from the label of text: the feature vector (or the characteristic) of a text is no change when we change the candidate label for the text. As mentioned in the Introduction section, it may be better if the *feature vector* of a text counts also the relationship between the text and its assigned label.

## 3. Proposed Model.

This section presents our Label-Oriented (LO) model, including: presentation of notations, label-oriented features, the algorithm in training phase and testing phase.

### 3.1. Notations.

In order to make easy to follow this section, we make use of these following notations.

- $x$ is an extracted term of a text $t$ in a set of texts $T$. $l$ is a label (class) in a label set $L$.
- $tf(x,t)$ is the term frequency of the term $x$ in the document $t$.
- $T_l$, $T_{\neg l}$ are the set of texts which are assigned (unassigned, respectively) to the label $l$.
- $T_l^x$, $T_{\neg l}^x$ are the set of texts, which are assigned (unassigned, respectively) to the label $l$, in which, every text contains the term $x$.
- $n_t$ is the number of terms in the text $t$.
- The *frequency* of a term $x$ regarding a set of texts $T$ is defined by the following formula:

$$fre(x,T) = \begin{cases} 0 & \text{if } \mid T \mid = 0 \\ \dfrac{\mid \{t \in T : x \in t\} \mid}{\mid T \mid} & \text{otherwise} \end{cases} \tag{1}$$

### 3.2. Label-oriented features.

The main idea of this approach is to figure out the relationship, if it has, between each term $x$ to a label $l$ by dividing the training set into two sets of texts $T_l$, $T_{\neg l}$ by the label $l$. Obviously, if the texts which contain the term $x$ appear in $T_l$ more frequently than in $T_{\neg l}$, then the possibility that a text containing the term $x$ will belong to the set $T_l$ may be higher than the possibility that the text belongs to the set $T_{\neg l}$; and vice versa, if the frequency of texts containing the term $x$ in $T_l$ is lower than that in $T_{\neg l}$, then the possibility that a text containing the term $x$ will belong to the set $T_l$ may be lower than the possibility that the text belongs to the set $T_{\neg l}$. In order to take this relation into account, the *Label-Oriented (LO) score* of a term $x$ regarding a label $l$

is then defined by the following formula:

$$score_{LO}(x,l) = \begin{cases} 0 & \text{if } fre(x, T_l \cup T_{\neg l}) = 0 \\ \dfrac{2 * fre(x,T_l)}{fre(x,T_l) + fre(x,T_{\neg l})} & \text{otherwise} \end{cases} \quad (2)$$

This is the ratio of frequency of the texts containing $x$ in the set $T_l$ over the average frequency of the texts containing $x$ in the set $T_l$ and $T_{\neg l}$. The bigger this score, the higher the possibility that the text containing the term $x$ belongs to the label $l$.

The *Label-Unoriented (LU) score* of a term $x$ regarding a label $l$ is also defined by the following formula:

$$score_{LU}(x,l) = \begin{cases} 0 & \text{if } fre(x, T_l \cup T_{\neg l}) = 0 \\ \dfrac{2 * fre(x,T_{\neg l})}{fre(x,T_l) + fre(x,T_{\neg l})} & \text{otherwise} \end{cases} \quad (3)$$

This is the ratio of frequency of the texts containing $x$ in the set $T_{\neg l}$ over the average frequency of the texts containing $x$ in the set $T_l$ and $T_{\neg l}$. The bigger this score, the higher the possibility that the text containing the term $x$ does not belong to the label $l$.

Theoretically, the formulas of $score_{LO}(x,l)$ and $score_{LU}(x,l)$ could represent the fact that the higher the value of $fre(x,T)$, the higher the possibility that the text containing the term $x$ does not belong to the label $l$; and vice versa. However, there is a case in which, this idea is not completely respected. Let us consider the case of two terms $x_1$ and $x_2$ regarding the label $l$. $T_l$ and $T_{\neg l}$ have 100 texts each. In $T_l$, there are 100 texts which contain $x_1$, and only one text which contains $x_2$. In $T_{\neg l}$, there is no text which contains $x_1$, and no text which contains $x_2$. Obviously, the $score_{LO}(x_1,l)$ should be greater than $score_{LO}(x_2,l)$. However, in Formula (2), both the values of $score_{LO}(x_1,l)$ and $score_{LO}(x_2,l)$ are 2. The reason is that the $fre(x_1,T_{\neg l})$ and $fre(x_2,T_{\neg l})$ are zero in this case. This makes the value of $fre(x_1,T_{\neg l})$ and $fre(x_2,T_{\neg l})$ have no role in the formula of $score_{LO}(x_1,l)$ and $score_{LO}(x_2,l)$.

In order to avoid this case, we added a parameter, that is $\alpha$ *minimal appearance factor*, into Formula (1):

$$fre(x,T) = \begin{cases} 0 & \text{if } \mid T \mid = 0 \\ \dfrac{\max(\alpha; \mid \{t \in T : x \in t\} \mid)}{\mid T \mid} & \text{otherwise} \end{cases} \quad (4)$$

This means that, in the case there is no text in $T$ which contains $x$, we consider that there is $\alpha$ text which contains $x$, the value of $\alpha$ could be various from 0 to 1. For instance, in the case of $x_1$ and $x_2$, if we use $\alpha = 0.5$, then the $score_{LO}(x_1,l) = 2*1/(1+0.005) \sim 1.99$. Meanwhile the $score_{LO}(x_2,l) = 2*0.01/(0.01+0.005) \sim 1.33$, the idea of the model is still consistent. The choice of the $\alpha$ value should be based on experiment. This is a subject to test on each dataset.

Accordingly, the *final score* of a term $x$ regarding a label $l$ is then defined by the following formula:

$$score(x,l) = (score_{LO}(x,l))^{\beta} - (score_{LU}(x,l))^{\beta} \quad (5)$$

where $\beta$ is a *power factor* to distinguish the label-oriented and label-unoriented scores. The higher the value of $\beta$, the more significant the difference between the *oriented score* and *unoriented score*. This value could be detected in experiment because it may depend on datasets.

The value of $score(x,l) > 0$ indicates that the term $x$ is more important to the texts in $T_l$ than the texts in $T_{\neg l}$; and vice versa, the value of $score(x,l) < 0$ indicates that the term $x$ is more important to the texts in $T_l$ than the texts in $T_{\neg l}$. In other words, if

$score(x, l) > 0$ then the possibility that the text containing the term $x$ belongs to the set $T_l$ will be higher than the possibility that text belongs to the set $T_{\neg l}$; and vice versa, if $score(x, l) < 0$ then the possibility that the text containing the term $x$ belongs to the set $T_l$ will be lower than the possibility that text belongs to the set $T_{\neg l}$.

The objective of the training phase is thus, to calculate the *label-oriented term score* of all terms in all texts in the training set.

3.3. **Label detection for a text.** In order to determine whether a new text $t$ could be assigned or unassigned to a label $l$, this model calculates the *label-oriented document score* of $t$. The *Label-Oriented Document – LOD score* of a text $t$ for label $l$ is then defined as follows:

$$score_{doc}(t, l) = \sum_{x \in t} tf(x, t) * score(x, l) \tag{6}$$

The value $score_{doc}(t, l) > 0$ implies that the text $t$ is closer to the texts in $T_l$ than the texts in $T_{\neg l}$ (The text $t$ thus may be assigned to the label $l$); and vice versa, if $score_{doc}(t, l) \leq 0$ then the text $t$ is closer to the texts in $T_{\neg l}$ than the texts in $T_l$ (The text $t$ should not be assigned to the label $l$).

3.4. **The algorithm.** The algorithm is composed of two main phases: *training* and *classifying*.

3.4.1. *Training phase.* The training phase's objective is to calculate the *label-oriented score* (Algorithm 1). In the first stage, the input texts are split into terms (n-grams) and the TF of each term is calculated within each text (lines 1-6). This is similar to the TF-IDF approach. In the second stage, the union $V$ of all terms is created (line 7). And then, the *label-oriented score* of each term $x \in V$ regarding each label $l \in L$ is calculated by formulas proposed in this model (lines 8-16). This is the main difference of the proposed model regarding the TF-IDF approach.

After this phase, each term $x \in V$ has a *label-oriented score* vector whose each element is a pair $< l, s >$: For each label $l \in L$, the term $x$ has a *label-oriented term score* $s$.

**Algorithm 1** Calculation of *label-oriented score*

**Input**: A set of text $T$, each text is assigned a label $l \in L$.
**Output**: The *label-oriented score* of all terms appeared in texts in $T$ for all labels $l \in L$.

1: **for all** text $t \in T$ **do**
2: Split $t$ into a set of term
3: **for all** term $x \in t$ **do**
4: Calculate the TF $tf(x, t)$
5: **end for**
6: **end for**
7: Create the union $V$ of all terms from the term set of all text $t \in T$
8: **for all** label $l \in L$ **do**
9: $T_l$ is the set of all texts which are assigned to the label $l$.
10: $T_{\neg l}$ is the set of all texts which are not assigned to the label $l$.
11: **for all** term $x \in V$ **do**
12: Calculate the *label-oriented score* $score_{LO}(x, l)$
13: Calculate the *label-unoriented score* $score_{LU}(x, l)$
14: Calculate the *final score* $score(x, l)$
15: **end for**
16: **end for**

3.4.2. *Classifying phase.* For a new text $t$, the choice of label to assign to the text is presented in Algorithm 2. Firstly, the text $t$ is split into terms and the TF of each term is calculated (lines 1-4). Then, instead of calculating the vector TF-IDF as the classical approach, the *label-oriented document score* of the text $t$ for each label $l \in L$ is calculated by formulas proposed in this model (lines 5-7). And then, the maximal score for all label $l \in L$ is detected (line 8). Finally, the label whose the *label-oriented document score* is maximal will be assigned to the text $t$ (lines 9-11).

**Algorithm 2** Detection of the label for a text

**Input**: The *label-oriented score* of all terms $x \in V$ for all labels $l \in L$. A text $t$.
**Output**: The most suitable label $l_t \in L$ for the text $t$.

1: Split $t$ into a set of term
2: **for all** term $x \in t$ **do**
3: Calculate the TF $tf(x, t)$
4: **end for**
5: **for all** label $l_i \in L$ **do**
6: Calculate the *label-oriented document score* – $score_{doc}(t, l_i)$
7: **end for**
8: Find the maximal value of $\max = \max\{score_{doc}(t, l_i)\}$ for all labels $l_i \in L$
9: **if** $\max > 0$ **then**
10: Return the label $l_i$ whose $score_{doc}(t, l_i) = \max$
11: **end if**
12: Return null, otherwise.

In the proposed model, there are two parameters which are needed to be detected during experiment: the *minimal appearance factor* $\alpha$ and the *power factor* $\beta$. The algorithm to detect the best value of these parameters is described in Algorithm 3 (This scenario is also used in the evaluation section to detect the best value of them for each dataset). As

**Algorithm 3** Detection of the best value of $\alpha$ and $\beta$ for a dataset

**Input**: A dataset.
**Output**: The best value of $\alpha$ and $\beta$.

1: Split the dataset into a training set and a testing set
2: Fix the value of $\alpha = 0.5$, and test the value of $\beta$
3: Start with $\beta = 1$
4: **repeat**
5: Run the training phase on the training set
6: Run the testing phase on the testing set
7: Measure the performance of the current $\beta$
8: Increase value of $\beta$ if the current performance is still better than the previous.
9: **until** The performance is no more increased
10: Fix the value of $\beta$ with the best value founded, and test the value of $\alpha$
11: **for** $\alpha$ from 0 to 1 (gradually increased by 0.1 or 0.05) **do**
12: Run the training phase on the training set
13: Run the testing phase on the testing set
14: Measure the performance of the current $\alpha$
15: Keep the best performance until current step.
16: **end for**
17: The best value of $\alpha$ is that whose performance is the best.

the value of $\alpha$ must be in the interval $[0, 1]$, meanwhile there is no upper bound value for $\beta$. Therefore, it could be better if we fixed the value of $\alpha$ at the middle of its interval to find the best value of $\beta$ (lines 2-10). And then, once the best value of $\beta$ is detected, it is fixed to test and find the best value of $\alpha$ (lines 11-17).

4. **Evaluation.** This section presents two kinds of experiment: First, the sensitive test to find the best parameters used in this model; Second, the comparison of the proposed model to some related works.

In all experiments, the considered models will be evaluated in two datasets:

- Short text classification: The chosen dataset is NRC Hashtag Emotion Lexicon (EmoLex), an association of words with six emotions (anger, fear, surprise, sadness, joy, and disgust) generated automatically from tweets with emotion-word hashtags [30].
- Regular text classification: This experiment uses the 20 Newsgroups dataset [23] as the input dataset. This dataset is widely used in the domain of machine learning and information retrieval. The main features of these two datasets are presented in Table 2.

Table 2. Comparison of two datasets features

| **Feature** | **EmoLex** | **20 Newsgroups** |
|---|---|---|
| Number of texts | 21000 | 20000 |
| Minimal text length (in word) | 2 | 75 |
| Maximal text length (in word) | 20 | 20000 |
| Average text length (in word) | 15 | 370 |
| Number of labels | 6 | 20 |

4.1. **Finding the best parameters of the model.** This section presents some experiments which are built to find the best value of the $\alpha$ minimal appearance factor and the $\beta$ power factor on the chosen datasets.

4.1.1. *Method.* Between two considered parameters to consider, while the value of $\beta$ may be various from 1 to unlimited, the value of $\alpha$ should be limited in from 0 to 1. Moreover, these two parameters are not independent. Therefore, we fixed the value of $\alpha = 0.5$ (the mean value in its interval) to find the best value of $\beta$, and then, fixed the best value of $\beta$ to find the best value of $\alpha$ on each considered dataset. Accordingly, we use theses following scenarios:

***Scenario 1: Test the effect of β on the EmoLex dataset***

1: For each label, select randomly 1500 texts whose label is that label, and 1500 other texts whose label is different from that label.
2: Use the k-fold cross-validation [5], divide this set into ten subsets (10-folds): each subset has about 300 texts, in which, 150 texts have the considered label, 150 remaining texts have other label.
3: Repeat from the first fold to the tenth fold:
  3.1: Select the $k^{th}$-fold, and consider it as the current *testing set* (of 300 texts).
  3.2: Group nine remaining folds into a set, and consider it as the current *training set* (of 2700 texts).

3.3: Run our model with fixed value of $\alpha = 0.5$ and several values of $\beta$: from 0 to 20[1].

3.4: Observe the two output parameters *accuracy* and *F1-score*. They are calculated based on the definition of Salton and McGill [37]:

* *Number of True Positive (TP)*: This is the number of texts which are assigned to the considered label. And in the results, it is also assigned to the same label.
* *Number of False Positive (FP)*: This is the number of texts which are NOT assigned to the considered label. However, in the results, it is assigned to the label.
* *Number of False Negative (FN)*: This is the number of texts which are assigned to the considered label. However, in the results, it is NOT assigned to the label.
* *Number of True Negative (TN)*: This is the number of texts which are NOT assigned to the considered label. And in the results, it is NOT assigned to the label.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} * 100\% \tag{7}$$

$$Precision = \frac{TP}{TP + FP} * 100\% \tag{8}$$

$$Recall = \frac{TP}{TP + FN} * 100\% \tag{9}$$

$$F1\text{-}score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

4: Repeat the steps 3.1 to 3.4 in ten times (10-folds) and then, take the *average value* of each *output parameter* for each considered value of $\beta$.

5: Repeat the steps 1 to 4 for all labels in the considered dataset, and then, take the *average value* of each *output parameter* for each considered value of $\beta$.

***Scenario 2: Test the effect of β on the 20 Newsgroups dataset.*** This is the same as the scenario 1, except that:

- This is applied to the 20 Newsgroups dataset with 20 labels.
- In the step 1, for each label, select randomly 1000 texts whose label is that label, and 1000 other texts whose label is different from that label.
- Therefore, in the step 2, each fold has about 200 texts, in which, 100 texts have the considered label, 100 remaining texts have other label.
- In the step 3.3, run the model with value of $\beta$ from 0 to 100.

***Scenario 3: Test the effect of α on the EmoLex dataset.*** This is the same as the scenario 1, except that:

- In the step 3.3, fixed the value of $\beta$ by the best value found in the scenario 1, run the model with value of $\alpha$: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.

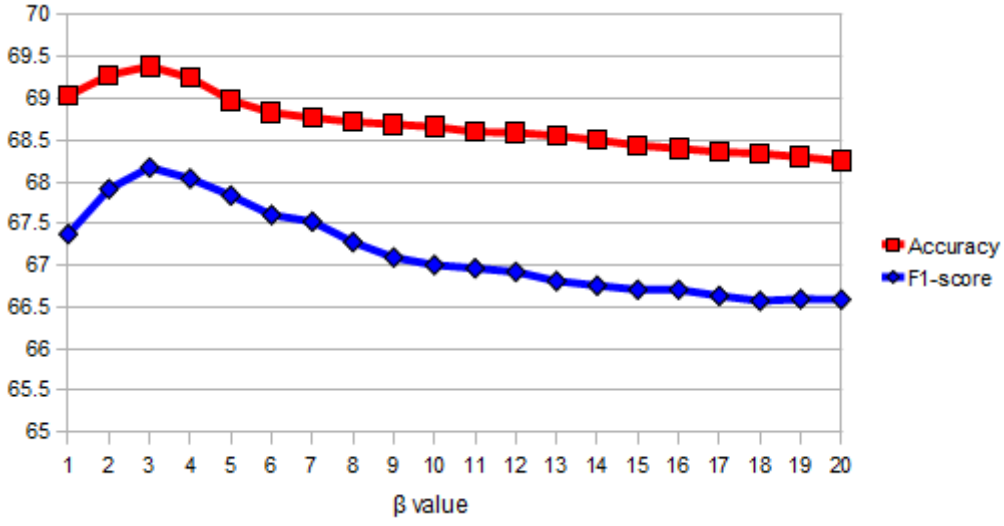
***Scenario 4: Test the effect of α on the 20 Newsgroups dataset.*** This is the same as the scenario 2, except that:

- In the step 3.3, fixed the value of $\beta$ by the best value found in the scenario 2, run the model with value of $\alpha$: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.

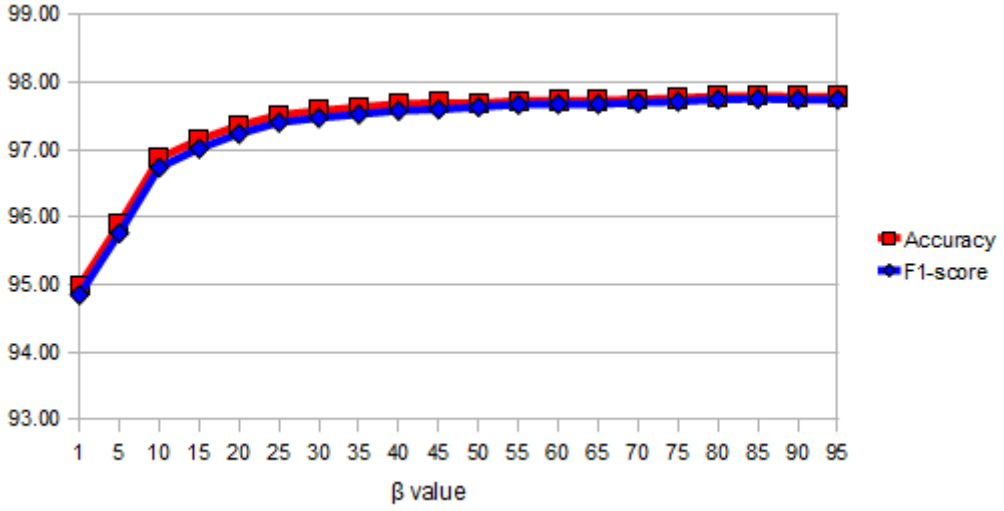
[1]In fact, after increasing the value of $\beta$ to 20 in this scenario, we recognized that the value of *accuracy* and *F1-score* are not better any more. Therefore, we do not test any more with the value bigger than 20. The same reason with the *scenario 2* where the maximal value of $\beta$ to test is 100.

4.1.2. *Results.* In the case of testing $\beta$ value, the results are presented in Figure 1. In it, the square line represents the values of *accuracy*, and the diamond line represents the values of *F1-score*. In the case of EmoLex dataset (Figure 1(a)), the results indicate that the values of *accuracy* and *F1-score* increase when the value of $\beta$ increases from 1 to 3. After that, both values are decreased when the value of $\beta$ increases. Therefore, we could consider the best value of $\beta$ for this dataset is 3. We will use this value for the next experiments on this dataset.

In the case of 20 Newsgroups dataset (Figure 1(b)), the results indicate that the values of *accuracy* and *F1-score* increase when the value of $\beta$ increases. However, when the value of $\beta$ is bigger than 50, the increment of *accuracy* and *F1-score* is no more significant. Consequently, we could consider the best value of $\beta$ for this dataset is about 50. We will use this value for the next experiments on this dataset.

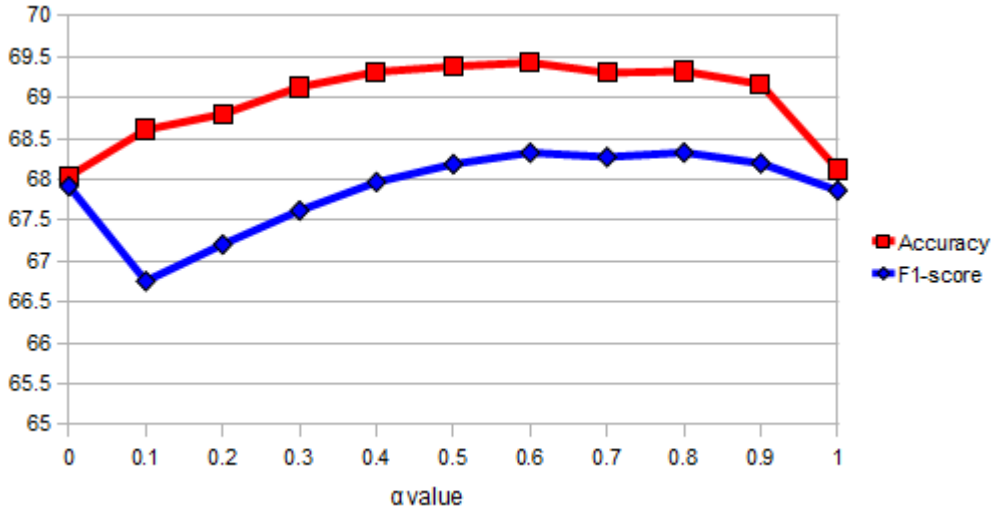



(a) EmoLex

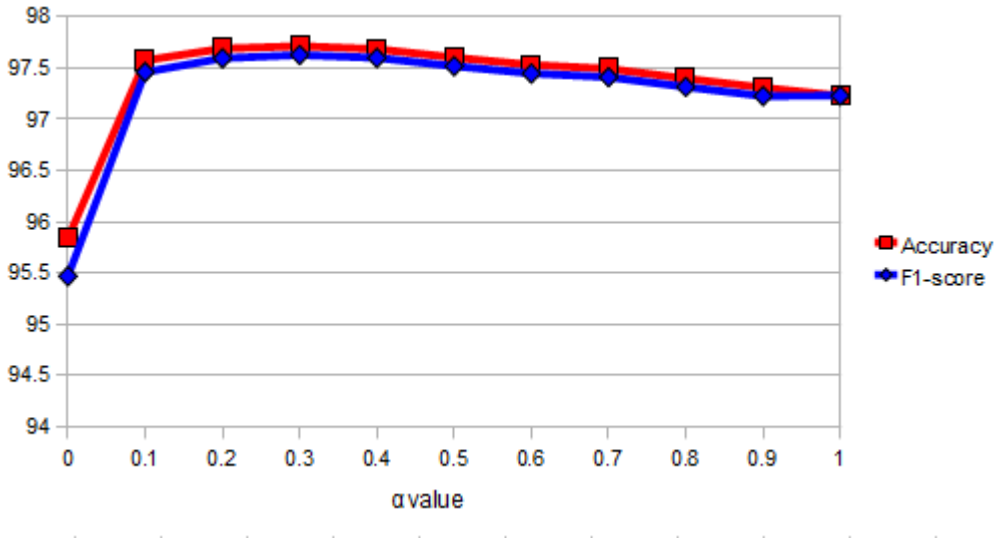



(b) 20 Newsgroups

FIGURE 1. Variation of accuracy and F1-score on the change of $\beta$ value




(a) EmoLex




(b) 20 Newsgroups

FIGURE 2. Variation of accuracy and F1-score on the change of $\alpha$ value

In the case of testing $\alpha$ value, the results are presented in Figure 2. In the case of EmoLex dataset (Figure 2(a)), the results indicate that the values of *accuracy* and *F1-score* increase when the value of $\alpha$ increases from 0.1 to 0.5. And then, both values are quite stable when $\alpha$ increases from 0.5 to 0.8. After that, both values are decreased when the value of $\alpha$ increases. Therefore, we could consider the best value of $\alpha$ for this dataset is in the interval $[0.5, 0.8]$. We will use $\alpha = 0.6$ for the next experiments on this dataset.

In the case of 20 Newsgroups dataset (Figure 2(b)), the results indicate that the values of *accuracy* and *F1-score* increase when the value of $\alpha$ increases from 0 to 0.3. After that, both values are decreased when the value of $\alpha$ increases. Therefore, we could consider the best value of $\alpha$ for this dataset is about 0.3. We will use $\alpha = 0.3$ for the next experiments on this dataset.

4.2. **Comparison to related works.** In this section, the proposed model will be compared to the following models:

- Model W2V (Word2Vec – Mikolov et al. [29]): This model is very closed to the proposed model in regarding the labeled and unlabeled texts for each label.
- Model based on TF-IDF: In this experiment, we choose the model FastText (FT) of Joulin et al. [20] to compare to our model.
- Model CNN (Convolution Neuron Network): This is recently one of the most popular and efficient models for pattern recognition/machine learning. There are several models proposed in this approach, such as model ELMo of Peters et al. [33], model BERT of Devlin et al. [12]. In this experiment, we choose the model BERT of Devlin et al. [12] to compare to our model.

4.2.1. *Method.* This experiment uses these following scenarios:

***Scenario 5: Comparison on the EmoLex dataset.*** This is the same as the scenario 1, except that:

- In the step 3.3, fixed value of $\alpha = 0.6$ and $\beta = 3$, then run four models: W2V, FT, BERT and our model (LO) in two phases: training with the *training set* and testing with the *testing set*.

***Scenario 6: Comparison on the 20 Newsgroups dataset.*** This is the same as the scenario 2, except that:

- In the step 3.3, fixed value of $\alpha = 0.3$ and $\beta = 50$, then run four models: W2V, FT, BERT and our model (LO) in two phases: training with the *training set* and testing with the *testing set*.

4.2.2. *Results.* In the case of EmoLex dataset, the results are presented in Table 3. In it, the first column is the name of six topics. The four next columns respectively represent the average value of *accuracy* of the four considered models, respectively. The last four columns also represent the value of *F1-score* of the four considered models, respectively. Each row represents the results of each topic. The last row represents the average value of all six topics of each model for the corresponding output parameter.

TABLE 3. Comparison of accuracy and F1-score (%) of W2V, FT, BERT and our model (LO) on EmoLex dataset

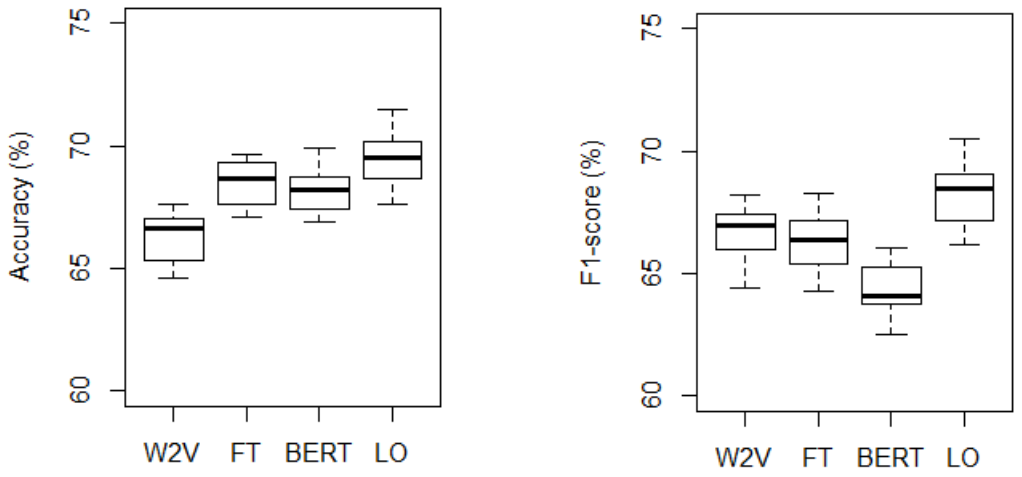
| **Emotion** | **Accuracy** | | | | **F1-score** | | | |
|---|---|---|---|---|---|---|---|---|
| | W2V | FT | BERT | LO | W2V | FT | BERT | LO |
| Joy | 69.62 | 69.80 | 69.21 | ***71.70*** | 72.30 | 69.04 | 65.66 | ***72.89*** |
| Sadness | 59.09 | 62.59 | 61.64 | ***63.84*** | ***66.87*** | 63.26 | 62.15 | 62.90 |
| Surprise | 67.39 | 65.61 | 65.70 | ***69.02*** | ***69.35*** | 64.85 | 65.66 | 68.66 |
| Fear | 69.47 | 72.30 | 70.46 | ***73.11*** | 65.14 | 72.09 | 69.04 | ***72.60*** |
| Disgust | 67.06 | 73.46 | ***75.48*** | 70.22 | 60.71 | 59.58 | 54.45 | ***63.08*** |
| Anger | 65.31 | 67.38 | 66.75 | ***68.69*** | 65.63 | 67.34 | 65.32 | ***69.84*** |
| **Average** | **66.32** | **68.52** | **68.21** | **69.43** | **66.67** | **66.29** | **64.28** | **68.33** |

The results indicate that the value of *accuracy* from our model (LO) is the highest on 5/6 topics, except the *disgust* is the highest in the model of BERT. Meanwhile, the model FT gets higher value than model BERT on 4/6 topics. In contrary, the model W2V gets the lowest value on 4/6 topics. Consequently, average value of *accuracy* is the lowest in the W2V model, the BERT model gets higher value than W2V model, the FT gets higher value than BERT model, and our model (LO) gets a little bit higher than FT (last row).

At the level of *F1-score* value, our model (LO) has the highest value on 4/6 topics. The model of W2V gets the highest value on two remaining topics (*sadness, surprise*). In contrary, the model BERT gets the lowest value on 4/6 topics. Consequently, average

value of *F1-score* is the lowest in the BERT model, the FT model gets higher value than BERT model, the W2V gets a bit higher value than FT model, and our model (LO) gets higher value than W2V (last row).

In order to see whether the difference is significant, we applied the *t-test* to our model values (for all six emotion labels) and those of W2V, FT, and BERT model respectively in two parameters: *accuracy* and *F1-score*. The statistical results are presented in Figure 3. At the level of *accuracy* (Figure 3(a)), the value of our model (69.43%) is significantly higher than that of W2V (66.32%) with the *p-value* $< 10^{-5}$. It is also significantly higher than that of BERT (68.21%) with the *p-value* $< 0.03$. However, it is not significantly higher than that of FT (68.52%) because the *p-value* $> 0.08$.




(a) Accuracy

(b) F1-score

FIGURE 3. Statistical comparison of accuracy and F1-score of W2V, FT, BERT and our model (LO) on EmoLex dataset

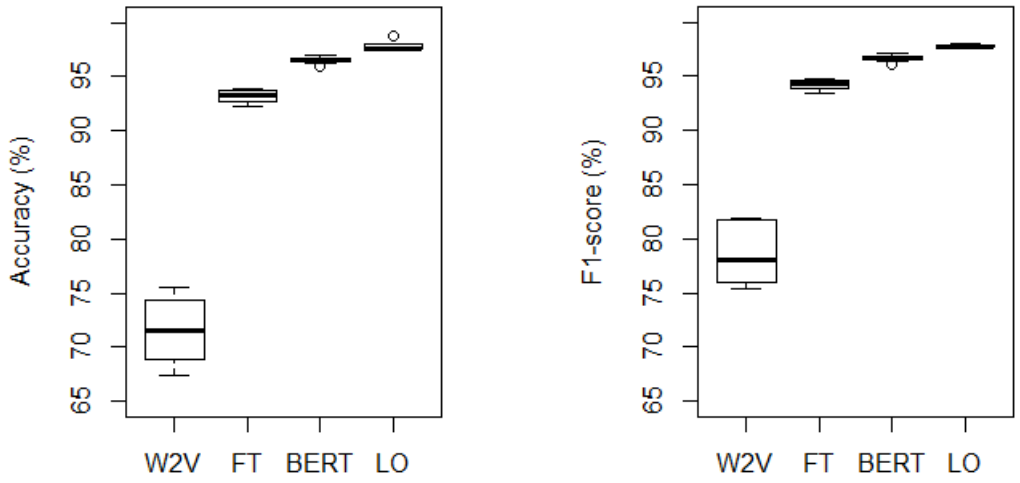
At the level of *F1-score* (Figure 3(b)), the value of our model (68.33%) is also significantly higher than that of W2V (66.67%) with the *p-value* $< 0.01$. It is also significantly higher than that of FT (66.29%) with the *p-value* $< 0.01$. It is also significantly higher than that of BERT (64.29%) with the *p-value* $< 10^{-5}$.

In the case of 20 Newsgroups dataset, the results are presented in Table 4. In it, the first column is the name of 20 topics. The four next columns respectively represent the average value of *accuracy* of the four considered models, respectively. The last four columns also represent the value of *F1-score* of the four considered models, respectively. Each row represents the results of each topic. The last row represents the average value of all 20 topics of each model for the corresponding output parameter. The results indicate that the value of *accuracy* from our model (LO) is the highest on 19/20 topics. Meanwhile, the BERT model gets the highest value of *accuracy* on the remaining topic (*talk.politics.mideast*). In almost topics, the W2V model gets the lowest value. The FT model gets higher value than W2V. The BERT gets higher value than FT. And our model (LO) gets higher value than BERT. Consequently, average value of *accuracy* is the lowest in the W2V model, the FT model gets higher value than W2V model, the BERT gets higher value than FT model, and our model (LO) gets the highest value (last row). The same results are obtained on the level of *F1-score* value.

In order to see whether the difference is significant, we also applied the *t-test* to our model values (for all 20 topics) and those of W2V, FT, and BERT model respectively in

TABLE 4. Comparison of accuracy and F1-score (%) of W2V, FT, BERT and our model (LO) on the 20 Newsgroups dataset

| Topics | Accuracy | | | | F1-score | | | |
|---|---|---|---|---|---|---|---|---|
| | W2V | FT | BERT | LO | W2V | FT | BERT | LO |
| alt.atheism | 77.14 | 95.45 | 96.00 | ***97.85*** | 83.03 | 96.09 | 96.12 | ***97.74*** |
| comp.graphics | 67.33 | 90.00 | 93.18 | ***96.26*** | 76.60 | 91.69 | 93.74 | ***96.17*** |
| comp.os...misc | 65.91 | 87.16 | 95.90 | ***97.80*** | 55.35 | 87.44 | 96.16 | ***97.71*** |
| comp.sys.ibm.pc.hardware | 71.25 | 87.73 | 95.53 | ***96.96*** | 79.99 | 90.16 | 95.63 | ***96.86*** |
| comp.sys.mac.hardware | 72.37 | 90.57 | 95.69 | ***97.71*** | 80.77 | 92.20 | 95.96 | ***97.60*** |
| comp.windows.x | 73.25 | 92.73 | 95.49 | ***97.71*** | 80.65 | 93.76 | 95.76 | ***98.67*** |
| misc.forsale | 76.25 | 91.14 | 96.92 | ***97.62*** | 83.12 | 92.59 | 97.08 | ***97.51*** |
| rec.autos | 75.91 | 93.86 | 97.13 | ***98.79*** | 78.30 | 94.78 | 97.26 | ***98.72*** |
| rec.motorcycles | 80.42 | 95.45 | 98.41 | ***99.53*** | 84.77 | 96.12 | 98.46 | ***99.50*** |
| rec.sport.baseball | 70.57 | 96.82 | 97.44 | ***99.16*** | 79.76 | 97.28 | 97.56 | ***99.11*** |
| rec.sport.hockey | 70.84 | 97.95 | 97.74 | ***98.41*** | 79.66 | 98.24 | 97.85 | ***98.33*** |
| sci.crypt | 65.11 | 94.43 | 98.00 | ***98.83*** | 76.57 | 95.30 | 98.03 | ***98.77*** |
| sci.electronics | 75.91 | 91.36 | 96.41 | ***97.57*** | 83.03 | 92.74 | 96.62 | ***97.46*** |
| sci.med | 63.64 | 93.30 | 98.21 | ***98.41*** | 75.68 | 94.34 | 98.25 | ***98.33*** |
| sci.space | 72.27 | 95.91 | 97.33 | ***98.41*** | 80.26 | 96.46 | 97.37 | ***98.33*** |
| soc.religion.christian | 62.00 | 98.07 | 99.33 | ***99.39*** | 74.95 | 98.33 | ***99.35*** | ***99.35*** |
| talk.politics.guns | 71.02 | 94.43 | 95.74 | ***97.24*** | 78.88 | 95.24 | 95.94 | ***97.14*** |
| talk.politics.mideast | 69.08 | 96.82 | ***98.41*** | 97.76 | 77.85 | 97.26 | ***98.45*** | 97.66 |
| talk.politics.misc | 72.16 | 87.61 | 94.21 | ***94.63*** | 80.25 | 90.12 | 94.46 | ***94.54*** |
| talk.religion.misc | 75.10 | 93.07 | 93.74 | ***96.03*** | 82.07 | 94.15 | 93.96 | ***95.90*** |
| **Average** | **71.38** | **93.19** | **96.53** | **97.80** | **78.58** | **94.21** | **96.70** | **97.77** |




(a) Accuracy

(b) F1-score

FIGURE 4. Statistical comparison of accuracy and F1-score of W2V, FT, BERT and our model (LO) on 20 Newsgroups dataset

two parameters: *accuracy* and *F1-score*. The statistical results are presented in Figure 4. At the level of *accuracy* (Figure 4(a)), the value of our model (97.80%) is significantly higher than that of W2V (71.38%) with the *p-value* $< 10^{-9}$. It is also significantly higher than that of FT (93.19%) with the *p-value* $< 10^{-8}$. It is also significantly higher than that of BERT (96.53%) with the *p-value* $< 10^{-7}$.

The same results at the level of *F1-score* (Figure 4(b)), the value of our model (97.77%) is also significantly higher than that of W2V (78.58%) with the *p-value* $< 10^{-9}$. It is also significantly higher than that of FT (94.21%) with the *p-value* $< 10^{-8}$. It is also significantly higher than that of BERT (96.70%) with the *p-value* $< 10^{-7}$. From these results, we could say that our model is also significantly better than the model of W2V, FT and BERT on the 20 Newsgroups dataset.

5. **Conclusion.** This paper presents a *Label-Oriented* (LO) model for text classification. The main contribution of this model is that it takes account of the relationship between a text and its assigned label by introducing the *label-oriented score* of each term to a label. This score represents the level of the importance of the term regarding all terms and texts assigned to the label compared to all terms and texts unassigned to the label. In the testing phase, the sum of this score of all terms in a text will help us to determine whether the text should be assigned to the label or not. The proposed model is then evaluated in both cases of short and regular texts. The experiment results indicate that the proposed model is significantly better than the baseline models on the used datasets.

Extending this model to apply to the problem of multi-label classification of texts is one of our perspectives in the near future.

## REFERENCES


[1] D. W. Aha, D. Kibler and M. K. Albert, Instance-based learning algorithms, *Machine Learning*, vol.6, pp.37-66, 1991.

[2] F. S. Al-Anzi, D. A. Zeina and S. Hasan, Utilizing standard deviation in text classification weighting schemes, *International Journal of Innovative Computing, Information and Control*, vol.13, no.4, pp.1385-1398, 2017.

[3] S. Albitar, S. Fournier and B. Espinasse, *An Effective TF/IDF-Based Text-to-Text Semantic Similarity Measure for Text Classification*, Springer International Publishing, Cham, 2014.

[4] R. Bekkerman, R. El-Yaniv, N. Tishby and Y. Winter, On feature distributional clustering for text categorization, *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, New York, NY, USA, pp.146-153, 2001.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.

[6] C. Bouras, V. Poulopoulos and V. Tsogkas, PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries, *Data & Knowledge Engineering*, vol.64, no.1, pp.330-345, 2008.

[7] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, vol.2, no.2, pp.121-167, 1998.

[8] G. Chen, D. Ye, Z. Xing, J. Chen and E. Cambria, Ensemble application of convolutional and recurrent neural networks for multi-label text categorization, *2017 International Joint Conference on Neural Networks (IJCNN 2017)*, Anchorage, AK, USA, pp.2377-2383, 2017.

[9] W. Dai, G.-R. Xue, Q. Yang and Y. Yu, Transferring Naive Bayes classifiers for text classification, *Proc. of the 22nd National Conference on Artificial Intelligence – Volume 1 (AAAI'07)*, pp.540-545, 2007.

[10] A. Danesh, B. Moshiri and O. Fatemi, Improve text classification accuracy based on classifier fusion methods, *2007 10th International Conference on Information Fusion*, pp.1-6, 2007.

[11] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese and C. Badue, Automated multi-label text categorization with VG-RAM weightless neural networks, *Neurocomputing*, vol.72, nos.10-12, pp.2209-2217, 2009.

[12] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *CoRR*, abs/1810.04805, 2018.
[13] G. Erkan, A. Hassan, Q. Diao and D. R. Radev, Improved nearest neighbor methods for text classification with language modeling and harmonic functions, *Technical Report, Technical Report CSE-TR-576-11*, Department of Electrical Engineering and Computer Science, University of Michigan, 2011.
[14] K. Glinka and D. Zakrzewska, *Effective Multi-label Classification Method for Multidimensional Datasets*, Springer International Publishing, Cham, 2016.
[15] A. Gliozzo, C. Strapparava and I. Dagan, Improving text categorization bootstrapping via unsupervised learning, *ACM Transactions on Speech and Language Processing*, vol.6, no.1, pp.1:1-1:24, 2009.
[16] S. L. Goldberg, D. Z. Wang and T. Kraska, CASTLE: Crowd-assisted system for text labeling and extraction, *The 1st AAAI Conference on Human Computation and Crowdsourcing*, 2013.
[17] Hendry and R.-C. Chen, Predicting business category with multi-label classification from user-item review and business data based on k-means, *ICIC Express Letters*, vol.13, no.3, pp.255-262, 2019.
[18] Hendry, R.-C. Chen, L.-H. Li and Q. Zhao, Using deep learning to learn user rating from user comments, *International Journal of Innovative Computing, Information and Control*, vol.14, no.3, pp.1141-1149, 2018.
[19] G. H. John and P. Langley, Estimating continuous distributions in Bayesian classifiers, *The 11th Conference on Uncertainty in Artificial Intelligence*, San Mateo, pp.338-345, 1995.
[20] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, Bag of tricks for efficient text classification, *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain, pp.427-431, 2017.
[21] A. M. Kibriya, E. Frank, B. Pfahringer and G. Holmes, Multinomial Naive Bayes for text categorization revisited, *Proc. of the 17th Australian Joint Conference on Advances in Artificial Intelligence (AI'04)*, Berlin, Heidelberg, pp.488-499, 2004.
[22] Y. Kim, Convolutional neural networks for sentence classification, *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1746-1751, 2014.
[23] K. Lang, NewsWeeder: Learning to filter netnews, *Proc. of the 12th International Conference on Machine Learning*, pp.331-339, 1995.
[24] J. H. Lau, D. Newman, S. Karimi and T. Baldwin, Best topic word selection for topic labelling, *Proc. of the 23rd International Conference on Computational Linguistics: Posters (COLING'10)*, Stroudsburg, PA, USA, pp.605-613, 2010.
[25] L. Li, H. Wang, X. Sun, B. Chang, S. Zhao and L. Sha, Multi-label text categorization with joint learning predictions-as-features method, *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal, pp.835-839, 2015.
[26] X. Li, N. Liu, C. Yao and F. Fan, Text similarity measurement with semantic analysis, *International Journal of Innovative Computing, Information and Control*, vol.13, no.5, pp.1693-1708, 2017.
[27] X. Li, C. Yao, Q. Zhang and G. Zhang, Semantic similarity modeling based on multi-granularity interaction matching, *International Journal of Innovative Computing, Information and Control*, vol.15, no.5, pp.1685-1700, 2019.
[28] B. Liu, Y. Dai, X. Li, W. S. Lee and P. S. Yu, Building text classifiers using positive and unlabeled examples, *Proc. of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, Washington, D.C., USA, pp.179-186, 2003.
[29] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *Proc. of the 26th International Conference on Neural Information Processing Systems – Volume 2 (NIPS'13)*, USA, pp.3111-3119, 2013.
[30] S. M. Mohammad and S. Kiritchenko, Using hashtags to capture fine emotion categories from tweets, *Computational Intelligence*, vol.31, no.2, pp.301-326, 2015.
[31] J. Nam, J. Kim, I. Gurevych and J. Fürnkranz, Large-scale multi-label text classification – Revisiting neural networks, *CoRR*, abs/1312.5419, 2013.
[32] K. Nigam, A. K. Mccallum, S. Thrun and T. Mitchell, Text classification from labeled and unlabeled documents using EM, *Machine Learning*, vol.39, nos.2-3, pp.103-134, 2000.
[33] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, Deep contextualized word representations, *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, LA, pp.2227-2237, 2018.

[34] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[35] J. Read, B. Pfahringer, G. Holmes and E. Frank, Classifier chains for multi-label classification, *Machine Learning*, vol.85, no.3, pp.333-359, 2011.
[36] R. G. Rossi, R. M. Marcacini and S. O. Rezende, Benchmarking text collections for classification and clustering tasks, *Technical Report 395*, Institute of Mathematics and Computer Sciences, University of Sao Paulo, 2013.
[37] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.
[38] D. Vilar, M. J. Castro and E. Sanchis, *Multi-Label Text Classification Using Multinomial Models*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
[39] T.-Y. Wang and H.-M. Chiang, Solving multi-label text categorization problem using support vector machine approach with membership function, *Neurocomputing*, vol.74, no.17, pp.3682-3689, 2011.
[40] Z. Wei, H. Zhang, Z. Zhang, W. Li and D. Miao, A Naive Bayesian multi-label classification algorithm with application to visualize text search results, *International Journal of Advanced Intelligence*, vol.3, no.2, pp.173-188, 2011.
[41] L. Yang, C. Li, Q. Ding and L. Li, Combining lexical and semantic features for short text classification, *Procedia Computer Science*, vol.22, pp.78-86, 2013.
[42] H.-F. Yu, P. Jain, P. Kar and I. Dhillon, Large-scale multi-label learning with missing labels, *Proc. of the 31st International Conference on Machine Learning, Volume 32 of Proceedings of Machine Learning Research*, Bejing, China, pp.593-601, 2014.
[43] Y. Yu, W. Pedrycz and D. Miao, Multi-label classification by exploiting label correlations, *Expert Syst. Appl.*, vol.41, no.6, pp.2989-3004, 2014.
[44] M.-L. Zhang, J. M. Peña and V. Robles, Feature selection for multi-label Naive Bayes classification, *Information Sciences*, vol.179, no.19, pp.3218-3229, 2009.
[45] M.-L. Zhang and Z.-H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, *IEEE Transaction on Knowledge and Data Engineering*, vol.18, no.10, pp.1338-1351, 2006.
[46] X. Zhang, J. Zhao and Y. LeCun, Character-level convolutional networks for text classification, *Proc. of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*, Cambridge, MA, USA, pp.649-657, 2015.
[47] P. Zhou and N. El-Gohary, Ontology-based multilabel text classification of construction regulatory documents, *Journal of Computing in Civil Engineering*, vol.30, no.4, 2016.