Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

# Social Media Toxic Content Filtering System using SOIR Model

**Nidhi Bhandari** *

*Corresponding Author, Department of Applied Mathematics, Indore Institute of Engineering and Technology, Indore, India. E-mail: nidhi.bhandari1@gmail.com

**Rachna Navalakhe**

Department of Applied Mathematics and Computational Science, Shri G. S. Institute of Technology and Science, Indore, India. E-mail: sgsits.rachna@gmail.com

**G.L Prajapati**

Department of Applied Mathematics, Indore Institute of Engineering and Technology, Indore, India. E-mail: glprajapati1@gmail.com

## Abstract

Social media is a popular data source in the research community. It provides different opportunities to design practical applications to favor humanity and society. A significant amount of people consumes social media content. Thus, sometimes content promoters and influencers publish misleading and toxic content. Therefore, this paper proposes an unhealthy content filtering system using the information retrieval model SOIR to identify and remove poisonous content from social media. The Semantic query Optimization-based Information Retrieval (SOIR) uses Fuzzy C Means (FCM) clustering to produce a particular data structure. To incorporate a query generation technique for the generation of multiple queries to increase the probability of correct outcomes. The SOIR model is modified in this work to utilize the model with the social media toxic content filtering model. The model uses linguistic and semantically information to craft new feature sets. The Part of Speech (POS) tagging is used to construct the linguistic feature. Finally, the pattern-matching algorithm is designed to classify the tweets as toxic or nontoxic. Based on lexical and semantic analysis of similar semantic queries (Tweets), it is identified with the class labels of the tweets. Twitter text posts are used to create training and test samples in this context. Here, a total of 2002 tweets are used for the experiment. The experimental study has been carried out with the different I.R. models (K-NN, Cosine) based on precision, recall, and F1-Score demonstrating the superiority of the proposed classification model.

**Keywords:** Text Mining; Semantic Knowledge; Information Retrieval; Sentiment Analysis; Lexical Pattern Analysis.

Journal of Information Technology Management, 2023, Vol. 15, Special Issue, pp. 78-94
Published by University of Tehran, Faculty of Management
doi: https://doi.org/ 10.22059/jitm.2023.91569
Article Type: Research Paper

Received: December 22, 2022
Received in revised form: January 15, 2023
Accepted: February 20, 2023
Published online: March 15, 2023

## Introduction

Information Retrieval (I.R.) is a technique to locate precise information in different data formats, i.e., text, image, video, and others. Among these data formats, the reader has a significant contribution. Thus, in the presented work, Text information retrieval is the critical study area. The text I.R. models use text mining techniques. Text mining techniques are data mining algorithms employed to recover user query relevance information (Agarwal, 2013). The I.R. model contains three key components: User query, Query processor, and generation of outcomes (Wang et al., 2017). However, the deficiency of these components can impact the performance of information retrievals, such as lack of user query keywords, inappropriate keyword selection, lack of similar data, ranking of results, and others (Bergamaschi et al., 2010).

In this context, an I.R. model has recently been presented to optimize user input queries for accurate content extraction with less time complexity. The model is named Semantic query Optimization-based Information Retrieval (SOIR). This I.R. model incorporates query optimization and an FCM clustering technique. The experimental results indicate the performance of the SOIR is better than previous models (Chalal, 2016). However, there are various classical application areas of the I.R. systems. Beyond these applications, the I.R. model can also be applied for pattern recognition. In this context, the proposed work is extended to harmful social media content filtering (Bohra et al., 2018).

The papers may be dissimilar from each other and may have different lengths and subjects of data. It may also consist of a significant amount of noise and unwanted content. Therefore, preprocessing is used to improve data quality and reduce noise. Two data preprocessing steps were adopted (1) Removal of stop words and (2) Removal of special characters. Thus, we have prepared two lists first contains the stop words (i.e. That, her, we), and the second list contains special characters (i.e., "," "@"). The algorithm replaces all the listed contents from the input documents. This paper proposes an extension of the SOIR model, which will be used for classifying the toxic contents of social media posts. This model is a promising technique for handling negative tweets from social media using lexical pattern analysis and semantical pattern analysis (Jianqiang & Xiaolin, 2017).

## Literature Review

The rising technology is responsible for improvements in the existing systems. This section provides a discussion about the SOIR model. The main aim is to improve the I.R. model in terms of considerable running time. Due to a large amount of text data in the database and the lack of practical techniques, a significant amount of time is required to locate the information (Pasquier et al., 2020). The large domain of documents (the documents available in the database) available is not necessarily similar in category and contents. The selection of query keywords is also not appropriate. Most of the time, users utilize irrelevant keywords to find the required information. Thus, we need to optimize search query keywords. However, the document length is not similar, and the contents are also not identical. The feature selection reduces data dimensions and speeds up the search process. It also reduces memory and time consumption. Thus, the Term Frequency – Inverted Document Frequency (TF-IDF) is used for document feature selection (Nafis & Awang, 2021).

The TF-IDF is used to compute the weight W of all the extracted features. Using W, the essential tokens for a document have been identified. But the tickets from the papers are different in length. Therefore, the fixed size of the feature vector is created and limited to 30 tokens as the maximum vector size (Onan & Toçoğlu, 2021). Further, the Fuzzy C Means (FCM) clustering is used to categorize training feature vectors in their subjective categories available in document storage (Fotovatikhah et al., 2018). Based on membership values, the partitions are made. After clustering, the training features are grouped to their content and similar subjects. The clustering results in a well-organized list of features which can be defined as given in Equation (1):

$$F = < F_n, k_{1,2,\ldots n}, C > \qquad (1)$$

Where F is the feature set, $F_n$ is the file name or index, $k_{1,2,\ldots n}$ is the list of keywords, and C is the class name or subject. The training feature vector F is stored in a database. This structured data feature is also helpful for efficient data retrieval. On the other hand, the SOIR system accepts the user query keywords to find the information. Therefore, the user query is transformed into a vector. User query Q can be a set of keywords as mentioned in Equation (2):

$$Q = \{q_1, q_2, \ldots, q_k\} \qquad (2)$$

A set of questions is prepared using synonyms to optimize the user query. Thus, an additional data table containing the keywords and relevant synonyms is ready (Yassine et al., 2022). This synonyms database is named $SDB_n$. In this algorithm, the query keywords are twisted multiple times to generate new queries using similar semantic words using $SDB_n$. The different search query increases the chances of finding accurate data. After generating multiple user queries, the search is performed. The search process is developed based on the k-Nearest Neighbour (k-NN) algorithm (Irfan et al., 2018; Matcha et al., 2019). The k-NN algorithm finds the distance between each query string and the training feature $F_o$. The distance between the query and data less than 0.25 is counted as the result. A comparative performance study has also been performed to justify SOIR with the Cosine similarity-based technique and k-NN-based I.R. model (Kreiss, & McGregor, 2019; Xu et al., 2015).

## Methodology

Text data is essential and can be used for communication with individuals and targeted audiences. In this context, private communication between individuals can be done through any messaging application. But when people want to target a significant amount of people to communicate something, they use public platforms to spare or publish the content. Moreover, not all the publishers of content are legitimately using social media. Therefore, the essential modifications are made based on user query optimization and domain categorization. In addition, a semantic data model is prepared to recognize similar semantic words to optimize the user query. The proposed SOIR model is demonstrated in two major parts, i.e., training and information retrieval. The SOIR technique is shown in Figure 1, which consists of the data stored in an unstructured format. This storage contains documents in raw form. After the search process, the records are produced from this storage.

Some nonsocial elements also utilize these platforms to execute propaganda, social hate, pornographic content, or misleading news. Therefore, to keep the social media surrounding clean, we need an accurate model to identify these kinds of posts or content from social media. The precision, Recall, and F-Score is calculated for this purpose. The measured mean performance parameters are visualized using a bar graph in Figure 2. Here the x-axis of this graph shows the parameter measured, and the y-axis shows the algorithms' precision, recall, and F-Score. Additionally, the detailed experimental observations with the different scenarios are given in Table 1. The SOIR is an extension of the recently introduced k-NN-based I.R. system. According to the results, the precision (accuracy) of the approach enhances the learning size of data, but SOIR provides more precise results.

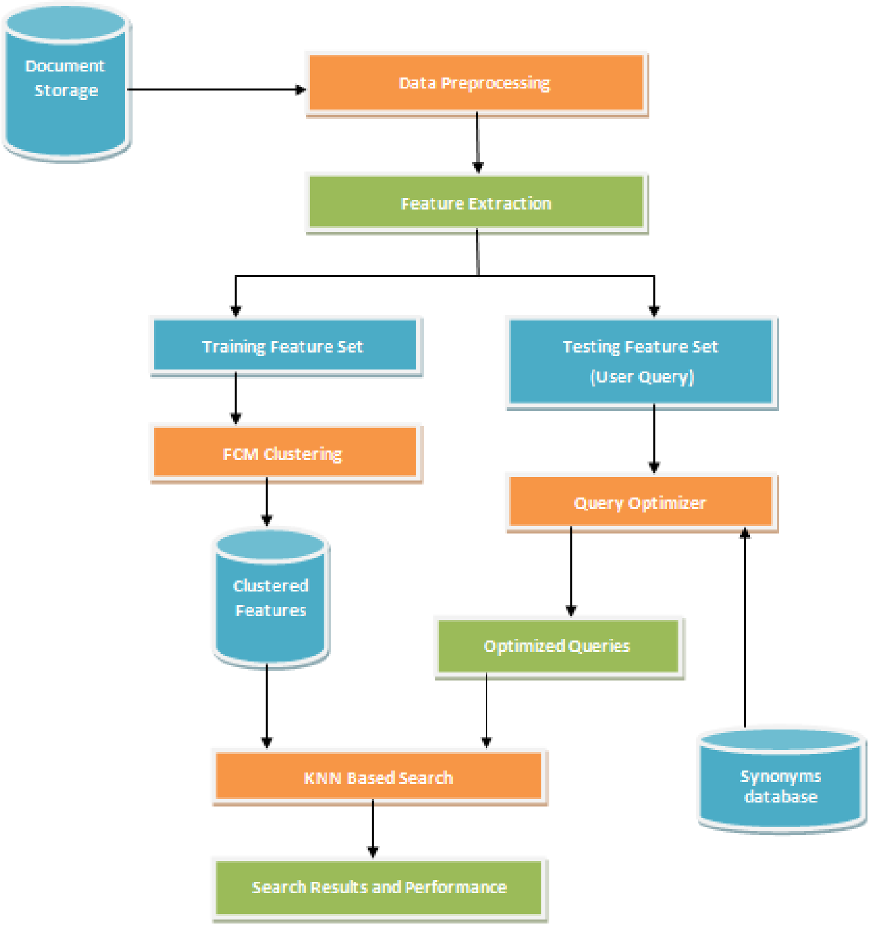


**Figure 1. Flowchart of the proposed SOIR model**

Similarly, the SOIR recall shows improved outcomes compared to previously offered techniques. Here for measuring performance, we also used F-score, which is used to represent the tread-off between precision and recall. According to the observed performance, the SOIR model performs much more accurately than our previously proposed model.

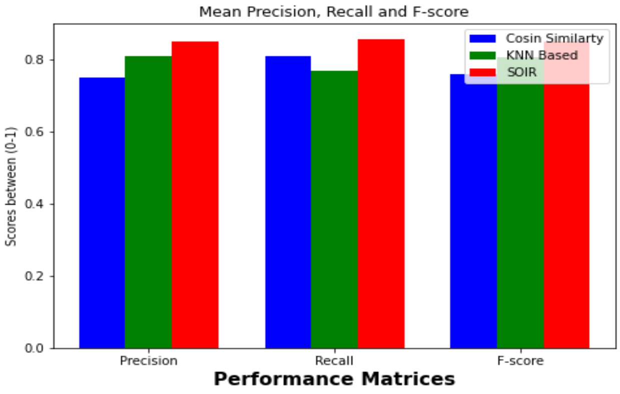


**Figure 2. Mean performance of SOIR**

**Table 1. Performance evaluation of SOIR-based technique**

| S. No. | Dataset size | Algorithms | Precision | Recall | F-Score |
|---|---|---|---|---|---|
| 1 | Data Mining (10)<br>Image Processing (10)<br>20 total | Cosine based | 0.6 | 0.73 | 0.646 |
| | | k-NN Based | 0.67 | 0.7 | 0.652 |
| | | SOIR | 0.72 | 0.74 | 0.729 |
| 2 | Data Mining (15)<br>Image Processing (15)<br>Big Data (10)<br>40 total | Cosine based | 0.68 | 0.77 | 0.704 |
| | | k-NN Based | 0.74 | 0.73 | 0.748 |
| | | SOIR | 0.78 | 0.79 | 0.784 |
| 3 | Data Mining (15)<br>Image Processing (15)<br>Big Data (15)<br>Cloud Computing (15)<br>60 total | Cosine based | 0.73 | 0.78 | 0.739 |
| | | k-NN Based | 0.78 | 0.75 | 0.775 |
| | | SOIR | 0.83 | 0.82 | 0.8249 |
| 4 | Data Mining (20)<br>Image Processing (20)<br>Big Data (20)<br>Cloud Computing (20)<br>Data security (20)<br>100 total | Cosine based | 0.77 | 0.83 | 0.774 |
| | | k-NN Based | 0.82 | 0.78 | 0.824 |
| | | SOIR | 0.87 | 0.86 | 0.8649 |
| 5 | Data Mining (30)<br>Image Processing (30)<br>Big Data (30)<br>Cloud Computing (30)<br>Data security (30)<br>150 total | Cosine based | 0.80 | 0.82 | 0.794 |
| | | k-NN Based | 0.88 | 0.79 | 0.847 |
| | | SOIR | 0.89 | 0.89 | 0.89 |
| 6 | Data Mining (50)<br>Image Processing (50)<br>Big Data (50)<br>Cloud Computing (50)<br>Data security (50)<br>250 total | Cosine based | 0.82 | 0.88 | 0.824 |
| | | k-NN Based | 0.90 | 0.83 | 0.887 |
| | | SOIR | 0.94 | 0.94 | 0.94 |
| 7 | Data Mining (100)<br>Image Processing (100)<br>Big Data (100)<br>Cloud Computing (100)<br>Data security (100)<br>500 total | Cosine based | 0.85 | 0.91 | 0.854 |
| | | k-NN Based | 0.92 | 0.86 | 0.912 |
| | | SOIR | 0.96 | 0.97 | 0.964 |

However, many of us formulate the toxic content classification problem in a supervised manner. Still, this work utilizes an information retrieval model for classifying social media content into regular or harmful class labels. This work is motivated by a recent contribution. The author provides a Hierarchical classification of emotional class labels. Using this classification, the category of a post into toxic or nontoxic contents of social media data needs to identify the negative emotions hidden in a social media post. The negative emotions are summarized in Table 2. The table consists of Emotion classes and the relevant flow of emotions with the associated emotional courses. According to the Table 2, we need to learn critical classes and their subclasses to identify the toxic contents of social media tweets.

**Table 2. Emotion classes**

| Emotions | Contains |
|---|---|
| Distressed | Sad, Disappointed, Guilty, Missed |
| Surprised | Surprised |
| Fearful | Panic, Frightened, Shy |
| Angry | Angry |
| Disgusted | Dissatisfied, Annoyed, Doubtful, Hateful |

In Table 2, there are mainly five emotional classes in negative subjects. Additionally, each negative course consists of its subclasses. Therefore, this is a multiclass classification problem in supervised learning, and to solve this problem by using the SOIR-based model. The Training process of the proposed model is given in Figure 3.

Training Samples: To train the proposed model, several sentiment-based tweeter datasets, but none of them contains the required classes and subclasses. Therefore, we have downloaded more than 3329 tweets from Twitter social media. Additionally, the tweets are categorized manually into their sentiment classes. Table 3 contains the emotional courses and the number of tweets available in each category of the training sample.

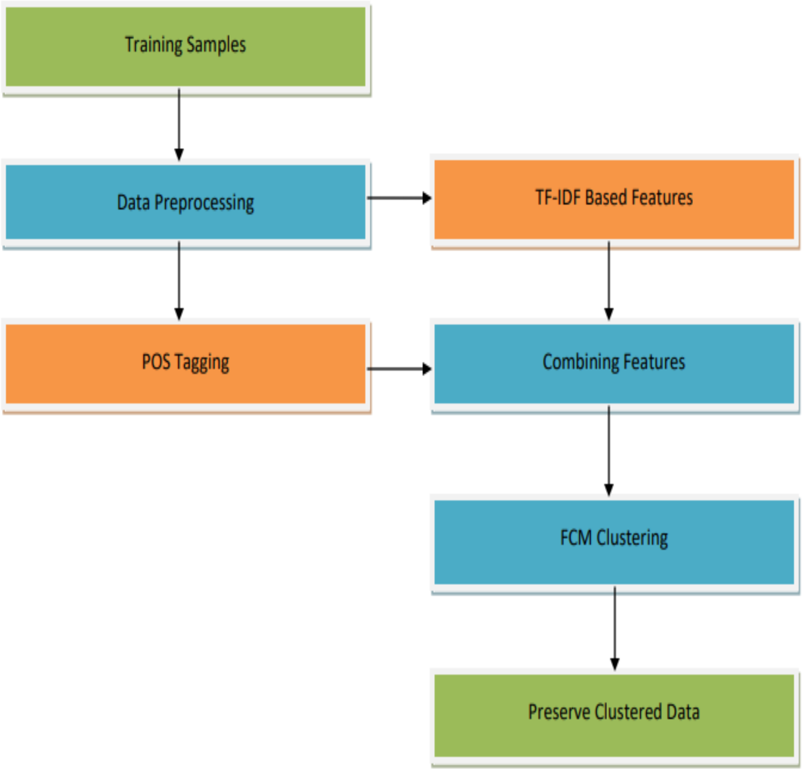


**Figure 3. The proposed training models**

**Table 3. Training samples**

| S. No. | Labels | No of tweets |
|---|---|---|
| 1 | Sad | 72 |
| 2 | Disappointed | 69 |
| 3 | Guilty | 108 |
| 4 | Missed | 67 |
| 5 | Surprised | 143 |
| 6 | Panic | 102 |
| 7 | Frightened | 127 |
| 8 | Shy | 151 |
| 9 | Angry | 207 |
| 10 | Dissatisfied | 159 |
| 11 | Annoyed | 197 |
| 12 | Doubtful | 247 |
| 13 | Hateful | 353 |
| **Total** | | **2002** |

**Data Preprocessing:** Preprocessing is essential in machine learning and data mining. The main aim of preprocessing is to enhance the information in the content and reduce the amount of noisy content. The following three preprocessing steps have been applied to clean the social media data.

- Remove tags and has tag-based content targets
- Remove other special characters
- Remove stop words

However, tweets on social media have a limited number of words (these contents are also known as micro-blogs), and the employment of preprocessing reduces the content significantly.

**POS Tagging:** POS Tagging is also known as part of speech tagging of the data. This may help us to understand the linguistic structure of the data. Based on the NLP, POS tags the fixed features prepared with their sentiment class labels.

**TF-IDF-based features:** Now, all the tweets are processed for measuring the TF-IDF. The TF-IDF is further converted into weights. The top 20 higher weighted tokens from the tweets are picked and used for representing the tweets. During this step, we found that some tweets that do not contain 20 keywords are also available. Thus, we extend the tweets using a temp string to complete their length.

**Table 4. NLP feature map**

| S. No. | Labels | NN | PRP | VB | ADV | ADJ | CC | POS |
|---|---|---|---|---|---|---|---|---|
| 1 | Sad | $T_{NN}^{Sad}$ | $T_{PRP}^{\text{Sad}}$ | $T_{VB}^{\text{Sad}}$ | $T_{ADV}^{\text{Sad}}$ | $T_{ADJ}^{\text{Sad}}$ | $T_{CC}^{\text{Sad}}$ | $T_{POS}^{\text{Sad}}$ |
| 2 | Disappointed | $T_{NN}^{\text{Disappointe}}$ | $T_{PRP}^{\text{Disappointed}}$ | $T_{VB}^{\text{Disappointe}}$ | $T_{ADV}^{\text{Disappointed}}$ | $T_{ADJ}^{\text{Disappointed}}$ | $T_{CC}^{\text{Disappointe}}$ | $T_{POS}^{\text{Disappointed}}$ |
| 3 | Guilty | $T_{\text{N.N.}}^{\text{Guilty}}$ | $T_{PRP}^{\text{Guilty}}$ | $T_{VB}^{\text{Guilty}}$ | $T_{ADV}^{\text{Guilty}}$ | $T_{ADJ}^{\text{Guilty}}$ | $T_{CC}^{\text{Guilty}}$ | $T_{POS}^{\text{Guilty}}$ |
| 4 | Missed | $T_{\text{N.N.}}^{\text{Missed}}$ | $T_{PRP}^{\text{Missed}}$ | $T_{VB}^{\text{Missed}}$ | $T_{ADV}^{\text{Missed}}$ | $T_{ADJ}^{\text{Missed}}$ | $T_{CC}^{\text{Missed}}$ | $T_{POS}^{\text{Missed}}$ |
| 5 | Surprised | $T_{\text{N.N.}}^{\text{Surprise}}$ | $T_{PRP}^{\text{Surprised}}$ | $T_{VB}^{\text{Surprised}}$ | $T_{ADV}^{\text{Surprised}}$ | $T_{ADJ}^{\text{Surprised}}$ | $T_{CC}^{\text{Surprised}}$ | $T_{POS}^{\text{Surprised}}$ |
| 6 | Panic | $T_{\text{N.N.}}^{\text{Panic}}$ | $T_{PRP}^{\text{Panic}}$ | $T_{VB}^{\text{Panic}}$ | $T_{ADV}^{\text{Panic}}$ | $T_{ADJ}^{\text{Panic}}$ | $T_{CC}^{\text{Panic}}$ | $T_{POS}^{\text{Panic}}$ |
| 7 | Frightened | $T_{\text{N.N.}}^{\text{Frightene}}$ | $T_{PRP}^{\text{Frightene}}$ | $T_{VB}^{\text{Frightene}}$ | $T_{ADV}^{\text{Frightene}}$ | $T_{ADJ}^{\text{Frightene}}$ | $T_{CC}^{\text{Frightene}}$ | $T_{POS}^{\text{Frightene}}$ |
| 8 | Shy | $T_{\text{NN}}^{\text{Shy}}$ | $T_{PRP}^{\text{Shy}}$ | $T_{VB}^{\text{Shy}}$ | $T_{ADV}^{\text{Shy}}$ | $T_{ADJ}^{\text{Shy}}$ | $T_{CC}^{\text{Shy}}$ | $T_{POS}^{\text{Shy}}$ |
| 9 | Angry | $T_{\text{N.N.}}^{\text{Angry}}$ | $T_{PRP}^{\text{Angry}}$ | $T_{VB}^{\text{Angry}}$ | $T_{ADV}^{\text{Angry}}$ | $T_{ADJ}^{\text{Angry}}$ | $T_{CC}^{\text{Angry}}$ | $T_{POS}^{\text{Angry}}$ |
| 10 | Dissatisfied | $T_{NN}^{\text{Dissatisfie}}$ | $T_{PRP}^{\text{Dissatisfi}}$ | $T_{VB}^{\text{Dissatisfie}}$ | $T_{ADV}^{\text{Dissatisfi}}$ | $T_{ADJ}^{\text{Dissatisfie}}$ | $T_{CC}^{\text{Dissatisfie}}$ | $T_{POS}^{\text{Dissatisfi}}$ |
| 11 | Annoyed | $T_{NN}^{\text{Annoyed}}$ | $T_{PRP}^{\text{Annoyed}}$ | $T_{VB}^{\text{Annoyed}}$ | $T_{ADV}^{\text{Annoyed}}$ | $T_{ADJ}^{\text{Annoyed}}$ | $T_{CC}^{\text{Annoyed}}$ | $T_{POS}^{\text{Annoyed}}$ |
| 12 | Doubtful | $T_{NN}^{\text{Doubtful}}$ | $T_{PRP}^{\text{Doubtful}}$ | $T_{VB}^{\text{Doubtful}}$ | $T_{ADV}^{\text{Doubtful}}$ | $T_{ADJ}^{\text{Doubtful}}$ | $T_{CC}^{\text{Doubtful}}$ | $T_{POS}^{\text{Doubtful}}$ |
| 13 | Hateful | $T_{NN}^{\text{Hateful}}$ | $T_{PRP}^{\text{Hateful}}$ | $T_{VB}^{\text{Hateful}}$ | $T_{ADV}^{\text{Hateful}}$ | $T_{ADJ}^{\text{Hateful}}$ | $T_{CC}^{\text{Hateful}}$ | $T_{POS}^{\text{Hateful}}$ |

**Combining features:** This method creates a threshold for identifying the different patterns. To prepare thresholds, we group the tweet's POS tags in their class labels. Here, 13 classes and 13 groups of tweets are used. Each tweet has been summarized in 7 features, as demonstrated in Table 4. Thus, seven different thresholds were created for each segment and each group. In the first step, compute the mean of the feature of the target group using the following Equation:

$$\mathrm{M_F} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{F_i} \tag{3}$$

Where $\mathrm{M_F}$ is the mean value of the particular feature, N total number of samples in the group.

After measuring the mean value, the distance from each point the group feature for measuring the limit is calculated using Equation (4).

$$\mathrm{L_F} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} |\mathrm{F_i} - \mathrm{M_F}| \tag{4}$$

Thus, the threshold of the particular feature of the above Equation can be modified as follows:

$$\mathrm{T_F}^{\mathrm{Group}} = \mathrm{M_F} \pm \mathrm{L_F} \tag{5}$$

Therefore, 13 groups and seven features are grouped from each group. Thus, 13*7 =91 parts have to be created, as demonstrated in Table 4. The feature map F.M. is used to classify actual tweets for identifying lexical information. To understand the threshold computation, compute the threshold for group sad and feature N.N. Thus, first, add all 72 instances of N.N. in Sad labelled data.

Further, 72 from the sum is divided into N.N. values. These results mean the value of the N.N. feature for the Sad group. Additionally, compute the mean of the difference from the mean value and return the threshold value after measuring the threshold for all the groups and features. Ninety-one feature maps are used to identify the harmful contents' linguistic structure. After that, a feature vector is created for the TF-IDF-based part extracted component.

FCM clustering: Here, FCM clustering helps us create the dictionary learning system. We are not using the entire FCM algorithm. We are just using the membership function of FCM for preparing the dictionary. The membership between data instance i and centroid j is measured using Equation (6):

$$\mu_{i,j} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{i,j}}{d_{i,k}}\right)^{\frac{2}{m-1}}} \quad (6)$$

The following algorithm is used for preparing the dictionary. According to the given process in able 5, the group-based or emotion label-based data is being processed. In this context, a random tweet is selected first from each group. This tweet is tokenized and inserted into the dictionary D with the TF-IDF weight. The TF-IDF can be measured using the below Equation:

$$W = TF * IDF \quad (7)$$

In the next step, one by one, each tweet in the group is taken and tokenized. Now, if a token exists in the dictionary, then we update the token's weight using the below-given Equation. And if the ticket is unavailable, we insert it in the dictionary with its associated TF-IDF weights. To compute the updated weight, the following Equation will be used:

$$NewWeight = 0.5 * Oldweight + 0.5 * \mu_{i,j} \quad (8)$$

The $\mu_{i,j}$ is the membership between the previous and new weight for the particular token.

**Preserved Clustered data:** The group-based prepared dictionaries are preserved in a database table for future use during classification. After completing the learning model, it is ready to classify the new tweets. Thus, a test dataset is also created. This test dataset is used for the validation of the model. To classify real-world tweets, design the following model for testing as given in Figure 4.

Test Samples: Some tweet samples need to be tested to validate this model. In this context, 50% of training samples and 50% of new tweets from Twitter have been used.

**Preprocessing Data:** The test samples are preprocessed in this step in a similar manner as described in the training set of the model.

**POS Tagging:** The similar NLP parser discussed in the training set has been used for tagging and preparing a set of 7 features.

**POS-Based Pattern Matching:** The tagged feature vector $T_v$ is used to compare with the thresholds as given in table 5. Additionally, the following algorithm matches the linguistic information pattern, as shown in Table 6.

**Table 5. Dictionary learning**

**Input:** label-based Grouped Data $G_n$
**Output:** dictionary features D

```
Process:
$for(i = 1; i < n; i++)$
    $Temp_m = ReadGroupData(G_i)$
    $for(j = 0; j < m; j++)$
        $if(j == 0)$
            $R = Random(1, m)$
            $S = get(Temp_R)$
            $[Token_o, Weight_o] = S.Tokenize()$
            $for(k = 1; k < o; k++)$
                $D_i.Insert(Token_k, Weight_k)$
            $end\ for$
        Else
            $S = get(Temp_j)$
            $[Token_o, Weight_o] = S.Tokenize()$
            $for(k = 1; k < o; k++)$
                $if(D.contains(Token_k))$
                    $D_i.Update(Token_k, NewWeight_k)$
                $else$
                    $D_i.Insert(Token_k, Weight_k)$
                $end\ if$
            $end\ for$
        End if
    $end\ for$
$end\ for$
Return D
```

**Table 6. Pattern Matching Algorithm**

```
Input: tagged feature vector $T_v$, the feature map $FM$
Output: best-matched pattern B

Process:
for$(i = 1; i < FM.rowCount; i++)$
    $[M_i, L_i] = FM_i$
    for$(j = 0; j < FM.colCount; j++)$
        $L_{max} = M_{i,j} + L_{i,j}$
        $L_{min} = M_{i,j} - L_{i,j}$
        if$(T_v(i,j) > L_{min}$ && $L_{max} \leq T_v(i,j))$
            $Match_i++$
        End if
    End for
End for
$H_{max} = getMaxVal(Match)$
Return $B = H_{max}.ClassLabel$
```

**POS-Based Decision:** The above-given algorithm is used for both of the things- pattern matching and decision making. Here the mean value $M$ of the threshold and limit $L$ are used for computing the upper threshold $L_{max}$and lower threshold $L_{min}$. The patterns between these limits compute the distance among queried tweets. POS tag feature and all the features threshold are given here as feature map F.M. Finally, the higher matched value-based class label is predicted as a decision.

**Tokenize tweet:** After lexical pattern-based decision-making, we use the semantics for categorizing a tweet as a final class label. Thus, the tweets are tokenized to get all the tokens in a tweet.

**Regenerate tweet:** As demonstrated in SOIR, a similar method is used to regenerate the tweets. Using the described query recreation process, we construct a different combination of keywords. Let us have a tweet such that:

$$TW = \{k_1, k_2, \ldots, k_n\} \tag{9}$$

After the processing regeneration process, the following Equation is obtained

$$TW_{m,n} = \begin{cases} k_{1,1}, k_{1,s}, \ldots\ldots\ldots\ldots., k_{1,n} \\ k_{2,1}, k_{2,2}, \ldots\ldots\ldots\ldots, k_{2,n} \\ k_{m,1}, k_{m,2}, \ldots\ldots\ldots\ldots, k_{m,n} \end{cases} \tag{10}$$

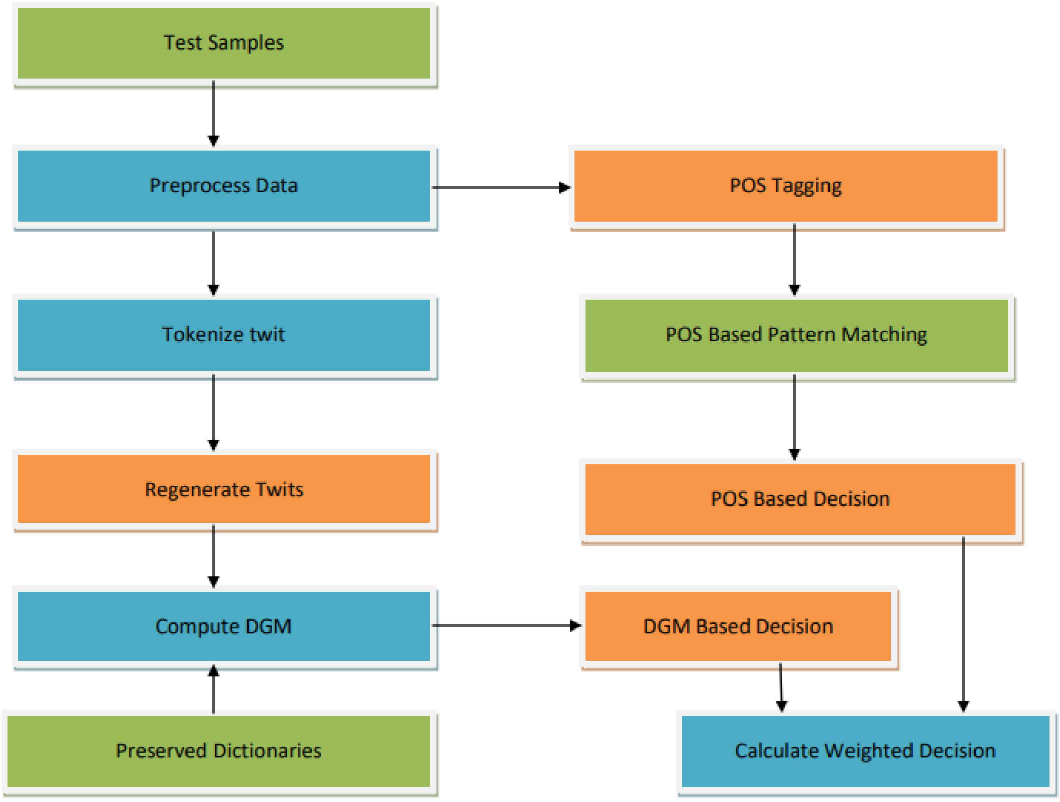


**Figure 4. Proposed classification system**

**Preserved Dictionaries:** After preparing the set of similar keyword tokens, we utilize the trained model, which contains the keywords and relevant weights for all the sentiment class labels. This can be defined as:

$$D_{o,p} = \begin{cases} D_1 = \{[T_{1,1}, W_{1,1}], [T_{1,2}, W_{1,2}] \ldots \ldots, [T_{1,p}, W_{1,p}]\} \\ D_2 = \{[T_{2,1}, W_{2,1}], [T_{2,2}, W_{2,2}] \ldots \ldots, [T_{2,p}, W_{2,p}]\} \\ D_o = \{[T_{o,1}, W_{o,1}], [T_{o,2}, W_{o,2}] \ldots \ldots, [T_{o,p}, W_{o,p}]\} \end{cases} \tag{11}$$

**Compute DGM:** This function generates a virtual directed graph to find a weight matrix used for decision-making. The matrix in the form of a graph model describes the association of a tweet with the given sentiment dictionary. In this context, an algorithm is developed to get the class label of the tweet, as shown in Table 7.

**Table 7. DGM-based classification**

**Input:** a set of semantically similar tweets $TW_{m,n}$, trained dictionary model $D_{o,p}$
**Output:** Class label C

```
Process:
1.      $for(i = 1; i < m; i++)$
a.      $for(j = 1; j < n; j++)$
i.      $if\left(D_i.contains(TW_{i,j})\right)$
1.      $W_i = W_{i-1} + W_{i,j}$
```

```
i.      $end\ if$
b.      $end\ for$
c.      $DW_i = \frac{W_i}{m}$
2.      End for
3.      $C = getMaxVal(DW_i).classlabel$
4.      Return C
```

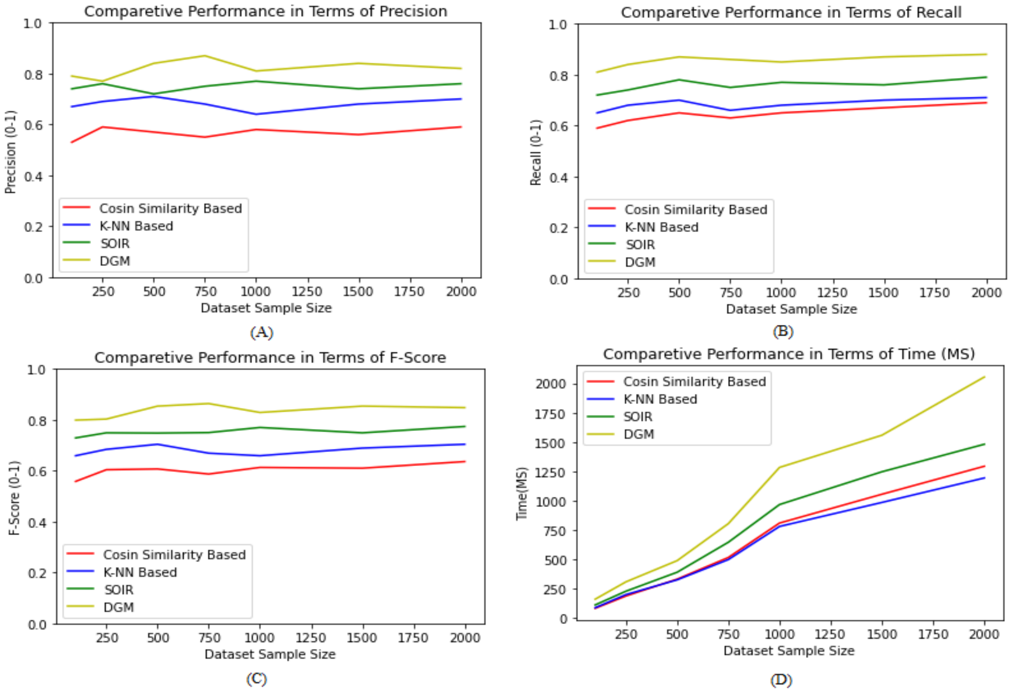
**DGM decision:** The above-given algorithm searches each word in the dictionary, and the relevant weights are aggregated for each dictionary. The more excellent value of these weights is used as the final sentiment label for the DGM model.

**Weighted decision:** Now, we have two decisions from two different approaches. To make a final decision, we provide a function that helps us decide.

$$f(B,C) = \begin{cases} B = C & then\ \ C \\ B! = C \qquad is\ B_{i-1} == C & then\ C \end{cases} \tag{12}$$

## Results and Discussion

Finally, Class label C belongs to shy, panic, sad, and guilty, then we label the tweet as nontoxic; if C returns other than these mentioned labels, the tweet is toxic. This section provides the formulation of the proposed social media toxic content filtering model using the SOIR extension. The following section provides a comparative performance study between the DGM-based and previously introduced models.



(A) (B) (C) (D)

**Figure 5. Comparative Performance Study of Implemented Techniques (A) Precision, (B) Recall, (C) F-Score, (D) Time consumption**

The proposed working model for information retrieval is evaluated in this section; the model classifies the social media text accurately for identifying the harmful content from the tweets using the Directional Graph Model-based Information Retrieval concept. Thus, different performance parameters are measured and reported in this section. Figure 5(A) shows the precision of all four techniques used to classify social media posts for increasing data. Different variants of datasets are prepared for training and testing. Precision indicates the accuracy of the pattern identification model. According to the results from the implemented techniques, the proposed Directed Graph Model (DGM) shows better accuracy than other given models. Similarly, the recall of all the methods has been measured and reported in Figure 5(B).

The given Figure demonstrates the performance of the implemented toxic content identification techniques. According to the obtained version of the model, we can see the proposed DGM-based approach performs more accurately than the other implemented models. But the conclusion can be made using the F1 score. Therefore, the proposed work also measures the performance of the models in terms of the F1-score. The F1 score of the techniques is given in Figure 5(C).

According to the performance of the models in terms of the F1-score, the SOIR and DGM technique provides more accurate results than other traditional I.R. models. However, the DGM performance is higher than SOIR, but the performance of SOIR is much more consistent than the DGM model. The following effective performance parameter is time consumption. The required time for training is measured and reported in Figure 5(D) for all the algorithms. The time requirements of the training algorithms have been measured in milliseconds. The x-axis of this diagram contains the training sample size, and the y-axis shows the time consumed. According to the measured results, the cosine-based and k-NN-based techniques are the winners. Additionally, the training of the other two models uses significant training time. But this model takes much less time to decide during the social media text classification.

## Conclusion

In this research, the information retrieval framework has been used to classify social media text. The proposed social media toxic content classification model extends the recently introduced I.R. model, namely SOIR. Thus, the paper includes the introduction of the SOIR model for retrieving text. Further, the methodology for developing the model is presented, which involves the Directed Graph Model (DGM) during the model's training. The main advantage of this model is that we can preserve the previously trained model for future use. After implementation, the model is compared with similar recently introduced models. The model's performance replicates the efficient and accurate modelling of identifying toxic tweets from social media. The proposed work will be extended further to improve the model's accuracy.

## Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

## Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article

## References

Agarwal, S. (2013, December). Data mining: Data mining concepts and techniques. In 2013 international conference on machine intelligence and research advancement (pp. 203-207). IEEE.

Bergamaschi, S., Domnori, E., Guerra, F., Orsini, M., Lado, R. T., & Velegrakis, Y. (2010). Keymantic: Semantic keyword-based searching in data integration systems. Proceedings of the VLDB Endowment, 3(1-2), 1637-1640.

Bohra, A., Vijay, D., Singh, V., Akhtar, S. S., & Shrivastava, M. (2018, June). A dataset of Hindi-English code-mixed social media text for hate speech detection. In Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media (pp. 36-41).

Chahal, M. (2016). Information retrieval using Jaccard similarity coefficient. Int. J. Comput. Trends Technol, 36, 140-143.

Fotovatikhah, F., Herrera, M., Shamshirband, S., Chau, K. W., Faizollahzadeh Ardabili, S., & Piran, M. J. (2018). Survey of computational intelligence as basis to big flood management: Challenges, research directions and future work. Engineering Applications of Computational Fluid Mechanics, 12(1), 411-437.

Irfan, M. R., Fauzi, M. A., Tibyani, T., & Mentari, N. D. (2018). Twitter sentiment analysis on 2013 curriculum using ensemble features and k-nearest neighbor. International Journal of Electrical and Computer Engineering (IJECE), 8(6), 5409-5414.

Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text preprocessing methods on twitter sentiment analysis. IEEE access, 5, 2870-2879.

Kreiss, D., & McGregor, S. C. (2019). The "arbiters of what our voters see": Facebook and Google's struggle with policy, process, and enforcement around political advertising. Political Communication, 36(4), 499-522.

Matcha, W., Gašević, D., Uzir, A., Jovanović, J., Pardo, A., Maldonado-Mahauad, J., & Pérez-Sanagustín, M. (2019, September). Detection of learning strategies: A comparison of process, sequence and network analytic approaches. In European conference on technology enhanced learning (pp. 525-540). Springer, Cham.

Nafis, N. S. M., & Awang, S. (2021). An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification. IEEE Access, 9, 52177-52192.

Onan, A., & Toçoğlu, M. A. (2021). Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts. Computer Applications in Engineering Education, 29(4), 675-689.

Pasquier, C., da Costa Pereira, C., & Tettamanzi, A. G. (2020, August). Extending a fuzzy polarity propagation method for multi-domain sentiment analysis with word embedding and pos tagging. In ECAI 2020-24th European Conference on Artificial Intelligence. 325, 2140-2147. IOS Press.

Wang, H., Zhang, Q., & Yuan, J. (2017). Semantically enhanced medical information retrieval system: a tensor factorization based approach. IEEE Access, 5, 7584-7593.

Xu, H., Yang, W., & Wang, J. (2015). Hierarchical emotion classification and emotion component analysis on Chinese micro-blog posts. Expert systems with applications, 42(22), 8745-8752.

Yassine, S., Kadry, S., & Sicilia, M. A. (2022). Detecting communities using social network analysis in online learning environments: Systematic literature review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12(1), e1431.

**Bibliographic information of this paper for citing:**

Nidhi, Bhandari; Rachna, Navalakhe & Prajapati, G.L. (2023). Social Media Toxic Content Filtering System using SOIR Model. *Journal of Information Technology Management,* 15 (Special Issue), 78-94. https://doi.org/ 10.22059/jitm.2023.91569