eTNT: Enhanced TextNetTopics with Filtered LDA Topics and Sequential Forward / Backward Topic Scoring Approaches

Daniel Voskergian[1], Rashid Jayousi[2], Burcu Bakir-Gungor[3]
Computer Engineering Department, Al-Quds University, Jerusalem, Palestine[1]
Computer Science Department, Al-Quds University, Jerusalem, Palestine[2]
Department of Computer Engineering-Faculty of Engineering, Abdullah Gul University, Kayseri, Turkey[3]

***Abstract*—TextNetTopics is a novel text classification-based topic modelling approach that focuses on topic selection rather than individual word selection to train a machine learning algorithm. However, one key limitation of TextNetTopics is its scoring component, which evaluates each topic in isolation and ranks them accordingly, ignoring the potential relationships between topics. In addition, the chosen topics may contain redundant or irrelevant features, potentially increasing the feature set size and introducing noise that can degrade the overall model performance. To address these limitations and improve the classification performance, this study introduces an enhancement to TextNetTopics. eTNT integrates two novel scoring approaches: Sequential Forward Topic Scoring (SFTS) and Sequential Backward Topic Scoring (SBTS), which consider topic interactions by assessing sets of topics simultaneously. Moreover, it incorporates a filtering component that aims to enhance topics' quality and discriminative power by removing non-informative features from each topic using Random Forest feature importance values. These integrations aim to streamline the topic selection process and enhance classifier efficiency for text classification. The results obtained from the WOS-5736, LitCovid, and MultiLabel datasets provide valuable insights into the superior effectiveness of eTNT compared to its counterpart, TextNetTopics.**

***Keywords*—Topic scoring; topic modeling, text classification; machine learning**

I. INTRODUCTION

In today's fast-paced information technology development, the volume of textual data is growing exponentially. This surge in unstructured and semi-structured content underscores the urgent requirement for effective methods to organize extensive amounts of information systematically. Text classification, which involves categorizing unlabeled text documents into predefined classes, is particularly challenging when dealing with the complexity of large datasets. Consequently, machine learning-based automatic text classification techniques are widely adopted across numerous applications [1].

However, automatic text classification tasks frequently deal with datasets encompassing tens of thousands of unique features, forming a high-dimensional challenge that can significantly impede the classification process. Despite the plethora of features, many do not significantly enhance the classification model; some may be less informative, introduce noise, or be redundant for predicting text labels. This situation results in longer computational times for training the learning algorithm, overfitting, substantial storage demands, and diminished classification performance and generalizability on test data. Hence, dimensionality reduction techniques, such as feature selection, are essential to mitigate these text classification issues [2].

The primary objective of feature selection methods is to determine a subset of features from the original set that accurately reflects the core information of the data. In text classification tasks, this subset is distinguished by its strong relevance to the class labels and ability to differentiate between classes effectively [3].

In this context, feature selection algorithms can be broadly classified into three categories: Filter methods utilize various metrics based on statistical principles and information theory to evaluate the correlation between features and class labels. These methods enable the ranking of features and the identification of the best subset according to predefined selection criteria. Wrapper methods determine the best subset by examining different combinations of features and evaluating their predictive power for the class labels using a specific classification algorithm. Lastly, Embedded methods select the optimal feature subset as part of the classifier training process [4].

Recently, a new direction has emerged that promotes the use of topic modeling (TM) as a novel method for feature reduction. In information retrieval, TM has proven to be a powerful tool due to its ability to identify latent themes, hidden variables, or abstract topics within a collection of documents without supervision. Each discovered topic represents a human-interpretable semantic notion. In addition to uncovering hidden structures in the data, topic modeling also offers a latent, interpretable representation of documents. Consequently, it serves as an automated method for understanding, organizing, and summarizing large volumes of text, enhancing the comprehension of the underlying themes in the data [5].

Topic modeling has been widely applied across various fields, significantly contributing to text classification as a feature projection method. In this area, topics derived from large text collections form features for representing documents [5]. Although topic modeling is commonly used for document

representation, its potential in feature selection has not been extensively explored in the existing research literature.

In this aspect, TextNetTopics [6] is a pioneering feature selection algorithm rooted in topic modeling and designed specifically for text classification. It employs Latent Dirichlet Allocation (LDA) as the foundational topic model to identify hidden topics, each comprising semantically related words that reflect the topic's theme. The algorithm utilizes a machine learning algorithm to evaluate the predictive performance of these topics (i.e., mean classification accuracy) and selects the top *r* topics with the highest discriminative power. These selected topics form a subset of words that effectively differentiate between two document classes in binary classification. This subset of topics is then used to train the classifier. An enhanced version called TextNetTopics Pro [7] has also been introduced, tailored explicitly for short-text classification.

The methodology of TextNetTopics is inspired by the G-S-M (Grouping, Scoring, and Modeling) approach [8], [9], initially used in the context of biological data. For a detailed examination of feature selection methods incorporating feature grouping, please consult the extensive review in [10].

However, a significant limitation of TextNetTopics is its reliance on ranking topics based solely on scores independently given to each topic, without accounting for the interactions and relationships among topics. Our research study presents innovative solutions to address this limitation and improve the effectiveness of topic scoring and ranking. By incorporating Sequential Topic Forward Scoring (STFS) and Sequential Topic Backward Scoring (STBS) into the TextNetTopics framework, we introduce a more refined and advanced topic selection process.

In addition, TextNetTopics treats a topic as a single entity to preserve the interaction between topic features. However, within these topics, some features may be less informative for the classification task, leading to an increase in the size of the final feature subset. In order to address this issue, this study introduces a filter component designed to remove the least important features from each topic. This filtering step aims to reduce redundancy and improve the overall relevance and efficiency of the feature set used for training a text classifier.

II. Related Work

Numerous research works have employed topic modeling as a method for feature projection [11], [12], [13]. This section focuses specifically on studies that use topic modeling as a technique for feature selection.

Zrigui et al. [14] utilized LDA to represent documents through real-valued features derived from terms associated with each topic. This method effectively reduces the dimensionality of the Vector Space Model (VSM) vectors while preserving both syntactic and semantic information in the document representation.

Zhang et al. [15] employed LDA with Gibbs Sampling for feature selection in text classification. They identified the most relevant terms within each topic by assessing term entropies in the term-topic matrix, selecting those with lower entropy values. These chosen features were then used to train a classifier. Their experiments showed that this method enhanced classification accuracy and reduced the dimensionality of the feature space.

Taşcı et al. [16] conducted similar research to [15], employing LDA for feature selection but utilizing Variational Expectation Maximization for estimation instead of Gibbs sampling. They compared their method with conventional feature selection techniques, including Chi-square, Information Gain, and Document Frequency. The results revealed that the LDA-based metrics performed comparably to those based on chi-square and document frequency.

Al-Salami et al. [17] applied a supervised variant of LDA, called Labeled LDA (LLDA), as a feature selection technique. LLDA restricts the number of topics to correspond with the number of categories in the corpus. The study selects words with high weights from LLDA's topic-word distribution matrix to train the classifier. The results revealed that using LLDA for feature selection improved the performance of AdaBoost with Multiclass Hamming Loss in multi-label categorization, outperforming other methods such as GSSC, Chi-square, and Information Gain.

Yousef and Voskergian [6] developed TextNetTopics, an LDA-based approach focusing on topic selection rather than individual word selection, where each topic represents a set of semantically related words. TextNetTopics aims to preserve the feature interactions within each topic by treating topics as single entities. Instead of utilizing all topics to train a classification model, TextNetTopics selectively chooses only the most relevant topics for training a machine learning algorithm for text classification. This approach acknowledges that some topics may introduce noise and potentially degrade the model's performance.

Voskergian et al. [7] introduced an enhanced version of TextNetTopics, called TextNetTopics Pro, explicitly designed for short text classification. This advanced approach utilizes a combination of word topics and topic distributions obtained from a short text topic model, addressing the issue of data sparsity commonly encountered in classifying short texts.

However, a notable limitation of TextNetTopics and its advanced version, TextNetTopics Pro, lies in the topic performance scoring (TPS) within the S component. TPS evaluates each topic separately without taking into account the impact of interactions and relationships between topics. This independent scoring can result in the selection of redundant topics or overlook topics that may be weak on their own but contribute significantly to performance when combined with other topics. This oversight highlights the need for topic-scoring refinement to improve overall classification performance.

Additionally, TextNetTopics treats topics as single entities, incorporating all topic features while training a classifier. This approach, however, can lead to the inclusion of irrelevant or redundant features within topics, which may increase the final feature subset size while diluting the classifier's discriminative power. Therefore, enhancing the quality of each topic in TextNetTopics by introducing a topic-feature filtering

component is essential for a more concise and discriminative topic selection process.

III. TOPIC PERFORMANCE SCORING (TPS)

The Topic Performance Scoring (TPS) method, introduced in [6], uses the training document-term dataset D_{train}, which includes d documents and t distinct terms, and the topic-term matrix TW, composed of k topics and m terms per topic to generate $d\times(m+1)$ dimensional topic-based sub-datasets D_{Ti}, each containing term features pertinent to a specific topic and the corresponding class label. A machine learning algorithm, such as Random Forest, is subsequently applied to each sub-dataset independently using the Monte Carlo cross-validation technique. Each topic receives a score based on a mean performance metric, such as mean accuracy or F1-score, resulting in a thorough evaluation of each topic's significance in the text classification task. This scoring approach is illustrated in Fig. 1.

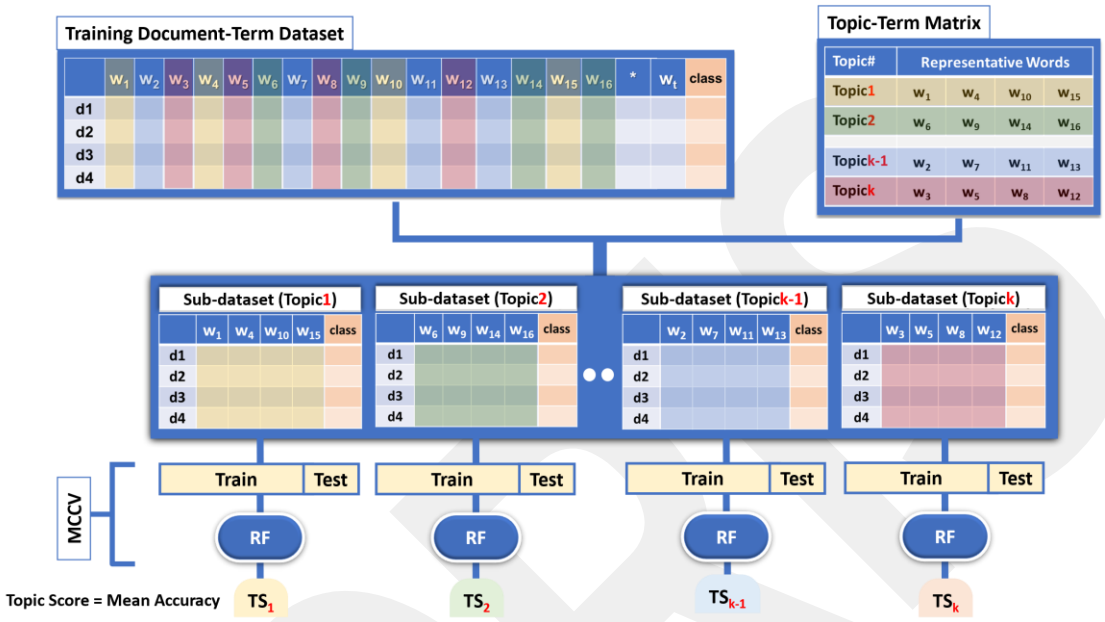

Fig. 1. The topic performance scoring approach.

IV. THE PROPOSED TOPIC-SCORING APPROACHES

This section presents two innovative scoring mechanisms, Sequential Forward Topic Scoring (SFTS) and Sequential Backward Topic Scoring (SBTS), developed to pinpoint the most significant topics for training machine learning algorithms. These mechanisms enhance the feature selection process by collaboratively assessing the importance of each topic and selecting the top-ranked ones that most effectively improve the model's predictive performance.

A. Sequential Forward Topic Scoring (SFTS)

This method performs scoring and ranking topics based on their contribution to the performance of an expanding list of topics. Let T be the set of k topics ($T = t_1, t_2, \ldots, t_k$). At iteration $r = 1$, the algorithm begins with an empty set of selected topics ($S_0 = \emptyset$) and iterates through each topic in T ($T = t_1, t_2, \ldots, t_k$), choosing the topic that results in the highest performance and adding it to the expanding set of selected topics (S_r). The considered performance metric is the mean F1-score of a Random Forest model, assessed through Monte Carlo cross-validation (RF_MCCV). However, one can select other metrics, such as accuracy or area under the curve. The number of internal iterations for the Monte Carlo cross-validation is specified by the user (e.g., 10).

During each subsequent iteration ($r = 2,3,\ldots,k$), the algorithm evaluates the potential performance of adding each topic t_i from the remaining topics ($T \backslash S_{r-1}$) to the current set of selected topics (S_{r-1}). The topic t_j that yields maximum performance is selected and incorporated into the expanding set (S_r). This topic t_j is then assigned a ranking index (R) corresponding to r, reflecting its importance and relevance to the classification task.

The iterative process continues until all k topics are included in the expanding set (S_r contains all k topics). Upon completion, the topics are ranked according to their assigned ranking indices ($R(t_j)$), with lower indices indicating topics of higher importance or relevance for the text classification task. Thus, the SFTS ranking reflects the sequence in which topics are added to the expanding topic set, with those added earlier being deemed more significant than those added later.

Algorithm 1 outlines the SFTS approach.

Algorithm 1: Sequential Forward Topic Scoring (SFTS)

Let $T = \{t_1, t_2, \ldots, t_k\}$ be the set of k topics (sets of related words).
Let S_r be the expanding set of topics at iteration r.
Let $P(.)$ be the performance metric (i.e., mean F1-score of RF_MCCV).
Let $R(.)$ be a function that assigns ranking indices to each topic.

Initialization: $S_0 = \emptyset$
For each iteration $r = 1,2,\ldots,k$:

(Evaluate the performance of adding each topic from T to the current set that is not already in S_{r-1})

$P(S_{r-1} \cup \{t_i\})$ for each $t_i \in T \backslash S_{r-1}$

(Select the topic t_j that upon inclusion, the set yields maximum performance)

$t_j = arg \max_{t_i} [P(S_{r-1} \cup \{t_i\})]$

(Update the expanding set)

$S_r = S_{r-1} \cup \{t_j\}$

(Assign a ranking index to the selected topic)

$R(t_j) = r$

End

The process continues until S_r includes all k topics. After completion, the topics are ranked based on their assigned ranking indices $R(t_j)$.

B. Sequential Backward Topic Selection (SBTS)

This method scores and ranks topics based on their impact on the performance of a reduced topic list. In iteration $r = 1$, the process begins with all k topics ($T = t_1, t_2, \ldots, t_k$) as the initial set ($S_0$). The algorithm then evaluates the potential performance of removing each topic (one at a time) from the current set of selected topics (S_{r-1}). The performance metric used is the mean F1-score of a Random Forest model, assessed through Monte Carlo cross-validation (RF_MCCV) with a user-defined number of internal iterations (e.g., 10). The topic whose removal results in the highest performance is chosen for permanent removal from the set and assigned a ranking index as $k + 1 - r$, where k represents the total number of topics and r is the current iteration number. The reduced topics set (S_r) is updated by excluding the selected topic.

This procedure continues until all topics have been removed from the reduced set ($S_k = \emptyset$). Finally, topics are ranked based on their assigned indices ($R(t_j)$), with lower indices indicating greater importance or relevance for the text classification task. Thus, the SBTS ranking reflects the sequence in which topics are removed from the reduced topic set, with those removed earlier being considered less important than those removed later.

Algorithm 2 outlines the SBTS approach.

Algorithm 2: Sequential Backward Topic Selection (SBTS)

Let $T = \{t_1, t_2, \ldots, t_k\}$ be the set of k topics (sets of related words).
Let S_r be the reducing set of topics at iteration r.
Let $P(.)$ be the performance metric (i.e., mean F1-score of RF_MCCV).
Let $R(.)$ be a function that assigns ranking indices to each topic.

Initialization: $S_0 = T$
For each iteration $r = 1,2, \ldots, k$:

(Evaluate the performance of removing each topic from S_{r-1})

$P(S_{r-1} \backslash \{t_i\})$ for each $t_i \in S_{r-1}$

(Select the topic t_j that upon removal, the set yields maximum performance)

$t_j = arg \max_{t_i} [P(S_{r-1} \backslash \{t_i\})]$

(Assign a ranking index to the selected topic)

$R(t_j) = (k + 1) - r$

(Update the reducing set)

$S_r = S_{r-1} \backslash \{t_j\}$

End

The process continues until S_r has no topics ($S_k = \emptyset$). After completion, the topics are ranked based on their assigned ranking indices $R(t_j)$.

V. Proposed Method: eTNT with Filtered LDA Topics and SFTS and SBTS Scoring Approaches

eTNT seeks to identify a concise subset of topics that maximizes discriminative power and relevance to class labels. The eTNT algorithm achieves this objective through five key components: T, F, G, S, and M.

The T component employs a Latent Dirichlet Allocation (LDA) topic model to uncover latent topics from a preprocessed document collection. Here, users need to define parameters such as the number of topics (*k*) and the number of terms per topic (*m*). This component primarily produces a topic-word matrix (*TW*) that details the association of words with each topic, each associated with specific probabilities.

The G component takes the topic-word matrix (*TW*) from the T component as input, along with the training Bag-of-Words (BOW) dataset (D_{train}). For each topic of *m* terms, the G component creates an *(m+1)*-dimensional sub-dataset from D_{train}, including the corresponding class label. Essentially, each sub-dataset represents a specific topic and includes mainly the words that coexist with that topic.

The F component trains a Random Forest (RF) model on each topic-based subdataset from the previous stage and extracts the importance values for each feature within the subdataset. To ensure robust and reliable feature importance evaluations, this process is repeated *z* times using a Monte Carlo Cross-Validation approach. In each iteration, a random selection of *b%* of the subdataset records is used to train the RF model. The feature importance values obtained from each iteration are then averaged, providing a stable and comprehensive assessment of the importance of each feature. This iterative process not only mitigates the risk of overfitting but also enhances the robustness of the feature importance values by accounting for variability within the data.

After ranking these feature importance values, a user-specified number *f* of highly ranked features (terms) are retained to represent the topic. Subsequently, new *f*-dimensional topic-based subdatasets are regenerated, each containing only the highly important features. This refinement enhances the quality of available topics, ensuring that they are more coherent and informative for subsequent classification

tasks, leading to potentially better model performance with a reduced feature set.

The S component utilizes SFTS and SBTS methods for scoring and ranking the refined topics with attention to topic interactions. These approaches assess and rank topics based on their impact on the performance of the expanding or reducing topic list. Performance is measured by the mean F1-score obtained through a Monte Carlo cross-validation process using a Random Forest model.

Sequential Forward Topic Scoring, or SFTS, starts with an initial empty set and evaluates each candidate topic via its corresponding two-class sub-dataset for performance. The topic demonstrating maximum performance is added to the expanding set as the top-ranked topic. The process then continues by adding the remaining $k-1$ topics to the current set, one at each time, evaluating the performance of their corresponding two-class sub-datasets, and including the topic that achieves the best performance in the growing set (this time, the topic is considered as a second-ranked topic). This iterative process continues until all topics or the desired number of topics are ranked. In this method, the last topic added receives the lowest rank.

Sequential Backward Topic Scoring, or SBTS, begins with a set of all topics and their corresponding two-class sub-dataset. It iteratively evaluates the impact of removing each topic from the set. The topic whose removal results in the highest performance is removed from the set and assigned the lowest rank. The process then continues with the current set of $k-1$ topics and its corresponding two-class sub-dataset, removing one topic at a time and assessing the effect on performance. The topic whose removal leads to the highest performance is excluded from the set and receives the second lowest rank. The iterative process continues until no topic remains, with the last topic removed being ranked the highest.

The M component aggregates the top-ranked topics incrementally in descending order, starting with the highest-ranked topic and progressively incorporating the remaining top-ranked topics until all desired topics are included. This process yields a combined set of terms for each topic aggregation and a corresponding two-class sub-datasets extracted from the training and testing of Bag-of-Words (BOW) datasets. The M component uses these sub-datasets to train and test a Random Forest model. It then identifies the optimal subset of topics, which results in the highest performance and discriminative ability for the text classification task. This optimal subset includes r topics, where r is less than k.

Fig. 2 and Fig. 3 illustrate the overall framework of eTNT, including the SFTS algorithm.

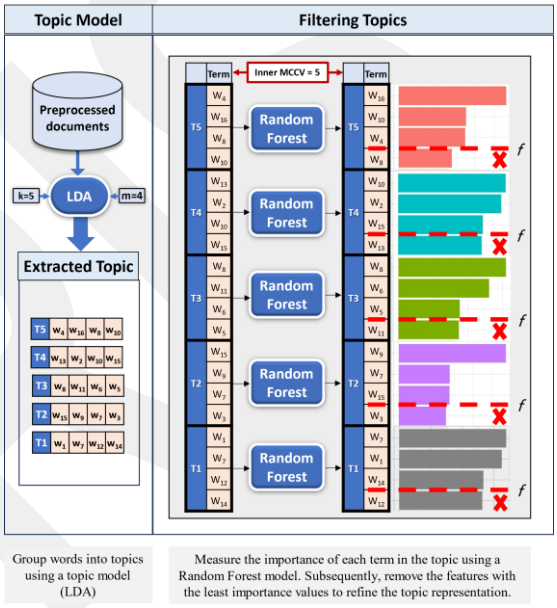

Fig. 2. The working mechanism of T and F components. k represents the number of topics, m referes to the number of terms in each topic, and f indicates the number of terms in each filtered topics.

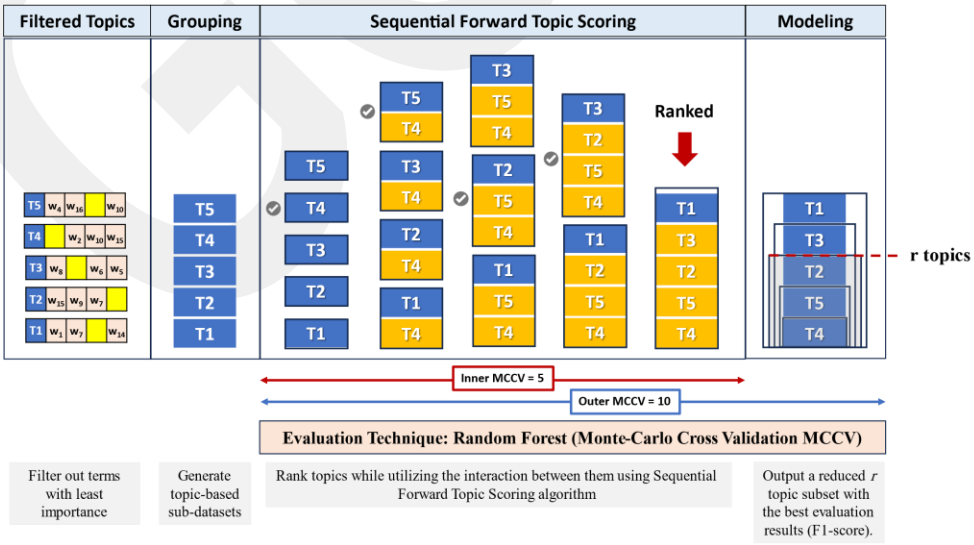

Fig. 3. The working mechanism of G, S and M components, using the SFTS approach.

VI. Experimental Work

A. Datasets

In this study, we utilized three datasets to assess the effectiveness of eTNT empirically:

The WOS-5736 dataset contains 5,736 documents classified into three higher-level classes. For the empirical evaluation of eTNT, we transformed the dataset into two balanced classes. We selected the largest category, with 2,847 abstracts, as the positive class, while the other classes (1,597 and 1,292 abstracts) formed the negative class [18].

The Multi-Label dataset contains 20,972 documents with abstracts and titles categorized under six labels: Quantitative Finance, Quantitative Biology, Computer Science, Statistics, Physics, and Mathematics. For eTNT evaluation, we selected 3,500 documents labeled Computer Science as the positive class and randomly sampled 3,500 documents without a Computer Science label to constitute the negative class instances [19].

The LitCovid dataset is a multi-label dataset with 16,127 records spanning five categories. This study focused on single-label records, resulting in the following distribution: 1,334, 1,632, 6,513, 119, and 429 for case reports, treatment, prevention, forecasting, and mechanism, respectively. To evaluate eTNT, we transformed the dataset into a binary class format by setting the prevention category as positive and the remaining categories as negative classes. We then performed dataset balancing by downsampling the positive class to 3,500 records to have a final dataset of 7,014 records [20].

B. Data Preprocessing

Text preprocessing is essential for improving the quality of analysis and reducing input data dimensionality, as raw documents often contain noisy and irrelevant data. In this study, KNIME workflows [21], [22] were executed to perform several Natural Language Processing (NLP) tasks.

These tasks include filtering out numbers, removing all punctuation, and excluding words with fewer than three characters. Additionally, all words are converted to lowercase to ensure uniformity. Stop words are eliminated using the Stop Word Filter node, and words are stemmed with the Snowball stemming library. Terms with a minimum document frequency below 1% are filtered out, and only English texts are retained.

C. Experimental Setup

We used KNIME workflows, available on KNIME Hub [22] and GitHub [21], to execute the natural language preprocessing tasks and evaluate TextNetTopics with the proposed eTNT. To extract latent topics, we employed the parallel thread implementation of LDA in KNIME [23]. We set the alpha parameter (Dirichlet prior on per-document topic distributions) and the beta parameter (Dirichlet prior on per-topic word distribution) to their default values of 0.1. The LDA training process was run for 1,000 iterations to estimate the topics. Based on performance, we chose 20 topics and 20 words per topic, as these settings consistently produced slightly better results than other configurations. In the filtering component, we selected ten terms from each topic since this approach yielded comparable performance to models with larger topics. This selection effectively balances model complexity and computational efficiency, ensuring optimal performance without unnecessary redundancy.

Regarding the experimental analysis, we used Monte Carlo stratified cross-validation (MCCV) with ten iterations to evaluate the model's performance more robustly. MCCV randomly partitions the original dataset into training and testing sets, with 90% allocated for training and 10% for testing in each iteration. The stratified approach maintains the class distribution, ensuring each subset maintains different classes' proportional representation from the original dataset, preventing bias in the evaluation process. The results from ten executed iterations are averaged to present the final performance.

D. Evaluation Measures

To evaluate the proposed eTNT's effectiveness compared to TextNetTopics, we perform a thorough analysis using various performance metrics, including Accuracy, Recall, Precision, and Area Under the Curve (AUC). The F1-Score is emphasized as a key metric for assessing classification performance.

VII. Experimental Results

A. Performance Evaluation of TextNetTopics Utilizing the Filtering Component

Tables I to III and Fig. 4 to Fig. 6 depict the performance of TextNetTopics with and without the proposed filtering component over different accumulated top-ranked topics. In this investigation, we used the default Topic Importance Scoring. According to the obtained results, TextNetTopics with the filtering component significantly outperforms the original TextNetTopics both in terms of classification performance and feature reduction. For instance, in the WOS-5736 dataset, it achieves a minimum and maximum F1-score of 89% and 97% using only 20 and 100 features, respectively, whereas TextNetTopics achieves an F1-score of 86% and 94% with 20 and 100 features. Similarly, in the MultiLabel dataset, it obtains a minimum and maximum F1-score of 78% and 84% using only 20 and 78 features, respectively, while TextNetTopics obtains an F1-score of 75% and 82% with 20 and 78 features. Likely, in the LitCovid dataset, it attains a minimum and maximum F1-score of 85% and 91% using only 20 and 125 features, respectively, while TextNetTopics attains an F1-score of 84% and 90% with 20 and 125 features. This improvement demonstrates the effectiveness of the filtering mechanisms in maintaining high classification accuracy with fewer feature set.

TABLE I. eTNT Performance Measures for the WOS-5736 Dataset

Topics	Terms	Accuracy	Recall	Specificity	F1	AUC	Precision	Cohen's kappa
TextNetTopics(TPS)	19.3	0.86	0.81	0.92	0.86	0.90	0.90	0.73
	28	0.87	0.83	0.91	0.86	0.91	0.90	0.74
	42.5	0.90	0.88	0.92	0.90	0.95	0.91	0.80
	56.2	0.93	0.94	0.93	0.93	0.97	0.93	0.86
	69	0.94	0.95	0.93	0.94	0.98	0.93	0.88

	79	0.94	0.95	0.93	0.94	0.98	0.93	0.88
	86.4	0.94	0.95	0.93	0.94	0.98	0.93	0.88
	93.4	0.94	0.95	0.94	0.94	0.99	0.94	0.89
	100	0.94	0.95	0.93	0.94	0.99	0.94	0.89
TextNetTopics(TPS) + F component	19.6	0.91	0.88	0.94	0.90	0.94	0.93	0.81
	25.6	0.92	0.91	0.94	0.92	0.96	0.94	0.85
	36.5	0.94	0.94	0.94	0.94	0.98	0.94	0.89
	43.6	0.95	0.95	0.94	0.95	0.98	0.94	0.89
	52.3	0.95	0.95	0.94	0.95	0.98	0.94	0.89
	61.9	0.95	0.96	0.95	0.95	0.99	0.95	0.91
	72.6	0.96	0.96	0.96	0.96	0.99	0.96	0.92
	80	0.96	0.97	0.96	0.96	0.99	0.96	0.92
	86	0.96	0.97	0.96	0.96	0.99	0.96	0.93
	100	0.97	0.97	0.96	0.97	0.99	0.96	0.93
eTNT(SFTS)	19	0.93	0.91	0.94	0.93	0.96	0.94	0.86
	34.9	0.95	0.95	0.95	0.95	0.98	0.95	0.90
	48.4	0.96	0.96	0.96	0.96	0.99	0.96	0.91
	58.2	0.96	0.96	0.96	0.96	0.99	0.96	0.92
	66.7	0.96	0.96	0.96	0.96	0.99	0.96	0.92
	73.7	0.96	0.96	0.96	0.96	0.99	0.96	0.92
	79.8	0.97	0.97	0.97	0.97	0.99	0.97	0.94
	86.5	0.97	0.97	0.97	0.97	0.99	0.97	0.93
	91.5	0.97	0.97	0.96	0.97	0.99	0.96	0.93
	100	0.97	0.97	0.96	0.97	0.99	0.96	0.93
eTNT(SBTS)	19.2	0.89	0.89	0.89	0.89	0.93	0.89	0.78
	35.9	0.94	0.95	0.94	0.94	0.98	0.94	0.89
	50.1	0.96	0.96	0.96	0.96	0.99	0.96	0.92
	59.8	0.96	0.97	0.96	0.96	0.99	0.96	0.92
	69.2	0.96	0.97	0.96	0.96	0.99	0.96	0.93
	77.2	0.97	0.97	0.96	0.97	0.99	0.96	0.93
	82.9	0.97	0.97	0.96	0.97	0.99	0.96	0.93
	87.9	0.97	0.97	0.97	0.97	0.99	0.97	0.94
	92.5	0.97	0.97	0.97	0.97	0.99	0.97	0.94
	100	0.97	0.97	0.97	0.97	1.00	0.97	0.94

TABLE II. eTNT PERFORMANCE MEASURES FOR THE MULTILABEL DATASET

Topics	Terms	Accuracy	Recall	Specificity	F1	AUC	Precision	Cohen's kappa
TextNetTopics(TPS)	20	0.74	0.75	0.74	0.75	0.80	0.74	0.49
	35	0.79	0.83	0.75	0.80	0.85	0.77	0.58
	45.8	0.79	0.84	0.75	0.80	0.86	0.77	0.59
	55.3	0.80	0.85	0.75	0.81	0.87	0.77	0.60
	64.8	0.80	0.86	0.75	0.81	0.88	0.77	0.60
	72.2	0.81	0.86	0.75	0.82	0.88	0.78	0.61
	80	0.81	0.87	0.75	0.82	0.89	0.78	0.62
TextNetTopics(TPS) + F component	21	0.78	0.82	0.75	0.78	0.85	0.77	0.57
	29.1	0.80	0.84	0.76	0.81	0.86	0.78	0.60
	38	0.81	0.86	0.76	0.82	0.88	0.78	0.62
	46	0.81	0.86	0.76	0.82	0.88	0.79	0.63
	54	0.81	0.87	0.76	0.82	0.89	0.78	0.63
	62	0.82	0.87	0.77	0.83	0.89	0.79	0.64
	67.8	0.82	0.88	0.77	0.83	0.90	0.79	0.64
	70	0.82	0.88	0.76	0.83	0.90	0.79	0.64
	72.8	0.83	0.88	0.77	0.83	0.90	0.79	0.65
	80	0.83	0.88	0.77	0.84	0.90	0.79	0.65
eTNT(SFTS)	20.5	0.78	0.81	0.75	0.79	0.84	0.77	0.57
	23.3	0.79	0.82	0.76	0.80	0.85	0.77	0.58
	27.9	0.80	0.84	0.76	0.81	0.86	0.78	0.60
	30.3	0.81	0.85	0.76	0.81	0.87	0.78	0.61
	38	0.82	0.87	0.76	0.83	0.88	0.79	0.63
	44.5	0.82	0.87	0.77	0.83	0.90	0.79	0.64
	52.3	0.82	0.87	0.78	0.83	0.90	0.80	0.65
	61.1	0.83	0.88	0.77	0.83	0.90	0.79	0.65
	69.1	0.83	0.88	0.77	0.84	0.90	0.80	0.65
	80	0.83	0.89	0.78	0.84	0.91	0.80	0.66
eTNT(SBTS)	19	0.76	0.77	0.75	0.76	0.82	0.75	0.52
	29.9	0.79	0.82	0.75	0.80	0.85	0.77	0.57
	35	0.80	0.84	0.75	0.81	0.87	0.78	0.60
	40	0.81	0.86	0.76	0.82	0.88	0.78	0.62
	50.3	0.82	0.87	0.77	0.83	0.90	0.79	0.64
	55.9	0.82	0.87	0.77	0.83	0.90	0.79	0.64

	61.2	0.83	0.88	0.78	0.84	0.90	0.80	0.65
	65.6	0.83	0.89	0.77	0.84	0.90	0.79	0.66
	71.5	0.83	0.89	0.77	0.84	0.91	0.79	0.66
	80	0.83	0.89	0.78	0.84	0.91	0.80	0.67

TABLE III. ETNT PERFORMANCE MEASURES FOR THE LITCOVID DATASET

Topics	Terms	Accuracy	Recall	Specificity	F1	AUC	Precision	Cohen's kappa
TextNetTopics(TPS)	20	0.84	0.85	0.83	0.84	0.91	0.83	0.68
	35	0.86	0.88	0.85	0.87	0.93	0.85	0.73
	48	0.87	0.88	0.86	0.87	0.94	0.86	0.74
	62.2	0.88	0.89	0.87	0.88	0.95	0.87	0.76
	73.6	0.89	0.89	0.88	0.89	0.95	0.88	0.78
	84.6	0.89	0.90	0.89	0.89	0.95	0.89	0.79
	101	0.89	0.90	0.89	0.89	0.96	0.89	0.79
	116	0.90	0.91	0.89	0.90	0.96	0.89	0.80
	128.3	0.90	0.91	0.89	0.90	0.96	0.89	0.80
	141.3	0.90	0.91	0.89	0.90	0.96	0.89	0.80
TextNetTopics(TPS) + F component	17	0.85	0.88	0.83	0.85	0.92	0.84	0.70
	27	0.87	0.89	0.85	0.87	0.94	0.86	0.74
	36	0.88	0.89	0.87	0.88	0.95	0.87	0.76
	50.3	0.89	0.90	0.88	0.89	0.96	0.88	0.78
	65.1	0.89	0.90	0.88	0.89	0.96	0.88	0.78
	77.7	0.90	0.91	0.89	0.90	0.96	0.89	0.80
	93.2	0.90	0.91	0.89	0.90	0.96	0.89	0.80
	109	0.90	0.91	0.89	0.90	0.96	0.89	0.80
	123	0.91	0.92	0.89	0.91	0.96	0.89	0.80
	141	0.90	0.92	0.89	0.90	0.96	0.89	0.80
eTNT(SFTS)	18.4	0.86	0.88	0.84	0.87	0.93	0.85	0.73
	30.8	0.88	0.89	0.88	0.88	0.95	0.88	0.77
	49.6	0.89	0.90	0.88	0.89	0.95	0.88	0.78
	66.3	0.89	0.90	0.89	0.89	0.96	0.89	0.79
	79.1	0.90	0.90	0.90	0.90	0.96	0.90	0.80
	92.4	0.90	0.91	0.90	0.90	0.96	0.90	0.81
	104.4	0.91	0.91	0.90	0.91	0.96	0.90	0.81
	117.6	0.91	0.92	0.90	0.91	0.97	0.90	0.82
	129.8	0.91	0.91	0.90	0.91	0.97	0.90	0.81
	141.5	0.91	0.92	0.90	0.91	0.97	0.90	0.82
eTNT(SBTS)	22	0.86	0.86	0.85	0.86	0.92	0.85	0.72
	39.2	0.89	0.89	0.88	0.89	0.95	0.88	0.77
	53.1	0.90	0.91	0.90	0.90	0.96	0.90	0.81
	68.6	0.90	0.91	0.89	0.90	0.96	0.90	0.81
	82	0.91	0.92	0.90	0.91	0.96	0.90	0.82
	94.7	0.91	0.91	0.90	0.91	0.96	0.90	0.81
	108.1	0.91	0.92	0.90	0.91	0.97	0.90	0.82
	123.1	0.91	0.92	0.90	0.91	0.97	0.91	0.82
	134.1	0.91	0.92	0.91	0.91	0.97	0.91	0.83
	142	0.91	0.92	0.90	0.91	0.97	0.91	0.83

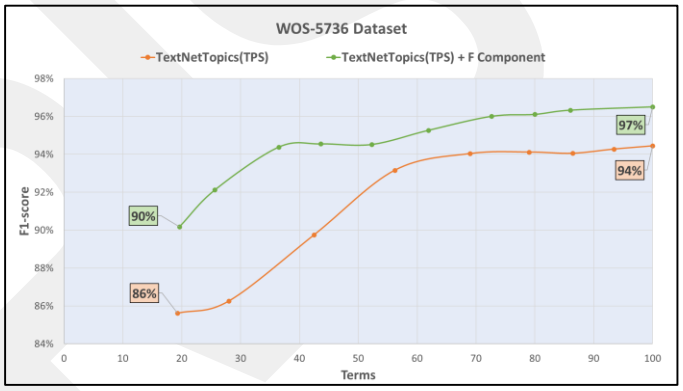

Fig. 4. F1-score performance comparison of TextNetTopics with and without the F component for the WOS-5736 dataset. The circles represent the number of top-ranked accumulated topics.

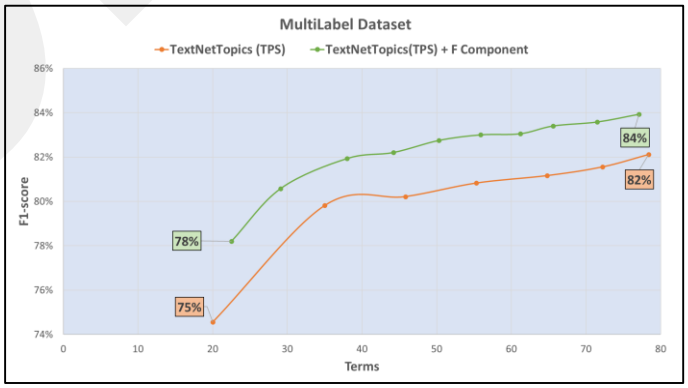

Fig. 5. F1-score performance comparison of TextNetTopics with and without the F component for the MultiLabel dataset. The circles represent the number of top-ranked accumulated topics.

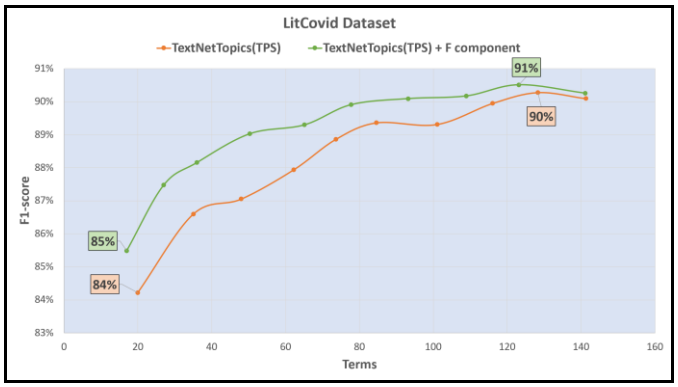

Fig. 6. F1-score performance comparison of TextNetTopics with and without the F component for the LitCovid dataset. The circles represent the number of top-ranked accumulated topics.

B. *eTNT Performance Using Various Topic Scoring Approaches*

In this subsection, we evaluate the performance of eTNT, which integrates the F component and one of the proposed scoring mechanisms—Sequential Forward Topic Scoring and Sequential Backward Topic Scoring—on three datasets: WOS-5736, LitCovid, and MultiLabel (refer to Fig. 7 and Fig. 9). Our analysis indicates a notable trend in the effectiveness of the examined approaches. We identified a pivotal turning point in the F1-score, marking a significant change in performance. The Sequential Forward approach showed an improved F1-score up to a specific number of topics. However, past this critical threshold, there was a shift, with the Sequential Backward approach becoming more effective.

This nuanced observation suggests that the optimal choice between forward and backward topic scoring depends on the number of topics involved. The backward approach proves to be more effective when a larger number of topics is necessary. In contrast, the forward approach is more advantageous when fewer topics are sufficient.

When comparing the original scoring mechanism, Topic Importance Scoring (TPS), with the two proposed approaches (refer to Fig. 7 till Fig. 9), an interesting trend emerges. Up to a specific number of topics, the original scoring mechanism outperformed the backward approach but lagged behind the forward approach. However, beyond a certain number of topics, the F1-score for TPS diminished compared to both the forward and backward approaches. This trend highlights the superiority of SFTS over TPS across all accumulated top-ranked topics.

According to the obtained results, SFTS and SBTS select a reduced number of features (accumulated topics) to achieve specific performance levels, thereby enabling further feature reduction. For instance, in the WOS-5736 dataset, to achieve a 97% F1-score, TPS utilizes 100 features, whereas SFTS and SBTS use only 80 and 77 features, respectively. Likely, in the MultiLabel dataset, to attain an 84% F1-score, TPS requires 78 features, while SFTS and SBTS use 70 and 61 features, respectively. Similarly, in the LitCovid dataset, to obtain a 91% F1-score, TPS utilizes 125 features, whereas SFTS and SBTS use only 104 and 82 features, respectively. These findings underscore the ability of SFTS and SBTS to enhance the topic selection process, improving efficiency without compromising classification performance.

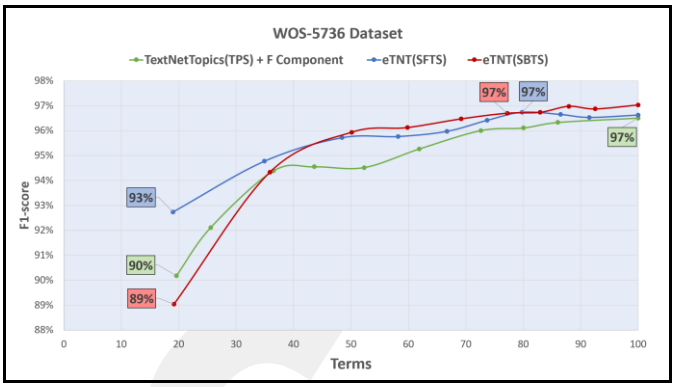

Fig. 7. F1-score performance comparison of eTNT with various topic scoring methods for the WOS-5736 dataset. The circles represent the number of top-ranked accumulated topics.

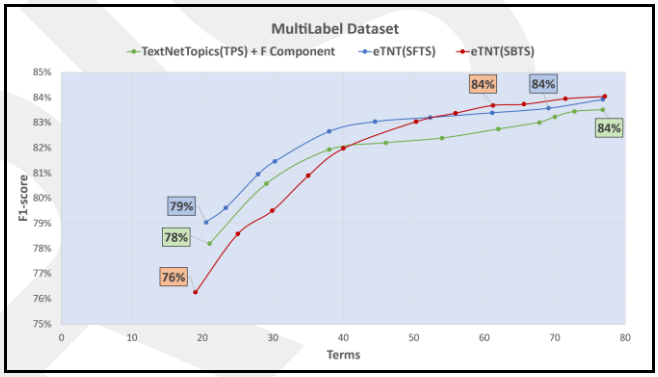

Fig. 8. F1-score performance comparison of eTNT with various topic scoring methods for the MultiLabel dataset. The circles represent the number of top-ranked accumulated topics.

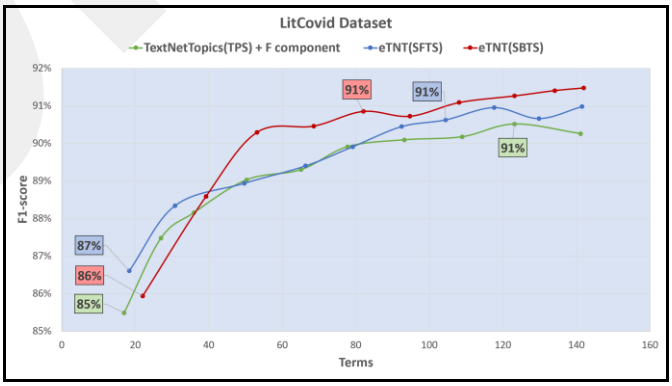

Fig. 9. F1-score performance comparison of eTNT with various topic scoring methods for the LitCovid dataset. The circles represent the number of top-ranked accumulated topics.

VIII. CONCLUSION

In conclusion, this study introduces eTNT, an enhancement of the TextNetTopics framework. eTNT integrates a filtering component that refines topic quality by removing non-informative features, thereby enhancing the informativeness and relevance of topics for text classification tasks. Additionally, it incorporates two novel scoring approaches: Sequential Forward Topic Scoring (SFTS) and Sequential

Backward Topic Scoring (SBTS). Unlike the original Topic Performance Scoring (TPS) method, which evaluates topics independently, SFTS and SBTS consider the interactions between topics, simultaneously assessing sets of topics to enhance the selection process and improve classifier efficiency.

The experimental results across the WOS-5736, LitCovid, and MultiLabel datasets provide valuable insights into the superior performance of eTNT over its predecessor, TextNetTopics. Specifically, eTNT demonstrates significant improvements in classification performance and feature reduction, underscoring the benefits of the proposed filtering and scoring mechanisms. For future work, we plan to investigate the use of word embeddings for feature grouping as an alternative to topic modeling in the T component, aiming to further enhance feature representation and classification performance.

ACKNOWLEDGMENT

We are so grateful to Prof. Malik Yousef for his significant support and expertise, which were crucial to accomplishing this study.

REFERENCES

[1] M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," Expert Systems with Applications, vol. 106, pp. 36–54, Sep. 2018.

[2] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," Multimed Tools Appl, vol. 78, no. 3, pp. 3797–3816, Feb. 2019, doi: 10.1007/s11042-018-6083-5.

[3] J. T. Pintas, L. A. F. Fernandes, and A. C. B. Garcia, "Feature selection methods for text classification: a systematic literature review," Artif Intell Rev, vol. 54, no. 8, pp. 6149–6200, Dec. 2021.

[4] E. O. Abiodun, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, and R. S. Alkhawaldeh, "A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities," Neural Comput & Applic, vol. 33, no. 22, pp. 15091–15118, Nov. 2021.

[5] A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat, and A. Hassan, "Topic modeling algorithms and applications: A survey," Information Systems, vol. 112, p. 102131, Feb. 2023, doi: 10.1016/j.is.2022.102131.

[6] M. Yousef and D. Voskergian, "TextNetTopics: Text Classification Based Word Grouping as Topics and Topics' Scoring," Front. Genet., vol. 13, p. 893378, Jun. 2022, doi: 10.3389/fgene.2022.893378.

[7] D. Voskergian, B. Bakir-Gungor, and M. Yousef, "TextNetTopics Pro, a topic model-based text classification for short text by integration of semantic and document-topic distribution information," Front. Genet., vol. 14, p. 1243874, Oct. 2023, doi: 10.3389/fgene.2023.1243874.

[8] M. Yousef, A. Kumar, and B. Bakir-Gungor, "Application of Biological Domain Knowledge Based Feature Selection on Gene Expression Data," Entropy, vol. 23, no. 1, p. 2, Dec. 2020, doi: 10.3390/e23010002.

[9] M. Yousef, J. Allmer, Y. İnal, and B. B. Gungor, "G-S-M: A Comprehensive Framework for Integrative Feature Selection in Omics Data Analysis and Beyond." Apr. 01, 2024. doi: 10.1101/2024.03.30.585514.

[10] C. Kuzudisli, B. Bakir-Gungor, N. Bulut, B. Qaqish, and M. Yousef, "Review of feature selection approaches based on grouping of features," PeerJ, vol. 11, p. e15666, Jul. 2023, doi: 10.7717/peerj.15666.

[11] L. Luo and L. Li, "Defining and Evaluating Classification Algorithm for High-Dimensional Data Based on Latent Topics," PLoS ONE, vol. 9, no. 1, p. e82119, Jan. 2014, doi: 10.1371/journal.pone.0082119.

[12] B. Al-Salemi, Mohd. J. Ab Aziz, and S. A. Noah, "LDA-AdaBoost.MH: Accelerated AdaBoost.MH based on latent Dirichlet allocation for text categorization," Journal of Information Science, vol. 41, no. 1, pp. 27–40, Feb. 2015, doi: 10.1177/0165551514551496.

[13] A. Glazkova, "Using topic modeling to improve the quality of age-based text classification," in CEUR Workshop Proceedings, 2021, pp. 92–97.

[14] M. Zrigui, R. Ayadi, M. Mars, and M. Maraoui, "Arabic Text Classification Framework Based on Latent Dirichlet Allocation," CIT, vol. 20, no. 2, 2012, doi: 10.2498/cit.1001770.

[15] Z. Zhang, X.-H. Phan, and S. Horiguchi, "An Efficient Feature Selection Using Hidden Topic in Text Categorization," in 22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008), Gino-wan, Okinawa, Japan: IEEE, 2008, pp. 1223–1228. doi: 10.1109/WAINA.2008.137.

[16] S. Tasci and T. Gungor, "LDA-based keyword selection in text categorization," in 2009 24th International Symposium on Computer and Information Sciences, Guzelyurt, Cyprus: IEEE, Sep. 2009, pp. 230–235. doi: 10.1109/ISCIS.2009.5291818.

[17] B. Al-Salemi, M. Ayob, S. A. M. Noah, and M. J. Ab Aziz, "Feature selection based on supervised topic modeling for boosting-based multi-label text categorization," in 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI), Langkawi: IEEE, Nov. 2017, pp. 1–6. doi: 10.1109/ICEEI.2017.8312411.

[18] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: Hierarchical Deep Learning for Text Classification," 2017, doi: 10.48550/ARXIV.1709.08267.

[19] "Multi-Label Classification Dataset." Accessed: Mar. 29, 2024. [Online]. Available: https://www.kaggle.com/datasets/shivanandmn/multilabel-classification-dataset

[20] "LitCovid dataset." Accessed: Oct. 29, 2023. [Online]. Available: https://drive.google.com/drive/folders/1mOmCy6mbBWXmfSzDyb6v4pG6pO-t_4At

[21] M. Yousef, "malikyousef/TextNetTopics-SFTS-SBTS." Mar. 17, 2024. Accessed: Mar. 22, 2024. [Online]. Available: https://github.com/malikyousef/TextNetTopics-SFTS-SBTS

[22] "malik/TextNetTopics-SFTS-SBTS," KNIME Community Hub. Accessed: Mar. 22, 2024. [Online]. Available: https://hub.knime.com/malik/spaces/TextNetTopics-SFTS-SBTS/

[23] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed algorithms for topic models.," Journal of Machine Learning Research, vol. 10, no. 8, 2009.