scientific reports

Check for updates

OPEN

Topic selection for text classification using ensemble topic modeling with grouping, scoring, and modeling approach

Daniel Voskergian[1]✉, Rashid Jayousi[2] & Malik Yousef[3]✉

TextNetTopics (Yousef et al. in Front Genet 13:893378, 2022. https://doi.org/10.3389/fgene.2022.893378) is a recently developed approach that performs text classification-based topics (a topic is a group of terms or words) extracted from a Latent Dirichlet Allocation topic modeling as features rather than individual words. Following this approach enables TextNetTopics to fulfill dimensionality reduction while preserving and embedding more thematic and semantic information into the text document representations. In this article, we introduced a novel approach, the Ensemble Topic Model for Topic Selection (ENTM-TS), an advancement of TextNetTopics. ENTM-TS integrates multiple topic models using the Grouping, Scoring, and Modeling approach, thereby mitigating the performance variability introduced by employing individual topic modeling methods within TextNetTopics. Additionally, we performed a thorough comparative study to evaluate TextNetTopics' performance using eleven state-of-the-art topic modeling algorithms. We used the extracted topics for each as input to the G component in the TextNetTopics tool to select the most compelling topic model regarding their predictive behavior for text classification. We conducted our comprehensive evaluation utilizing the Drug-Induced Liver Injury textual dataset from the CAMDA community and the WOS-5736 dataset. The experimental results show that the Latent Semantic Indexing provides comparable performance measures with fewer discriminative features when compared with other topic modeling methods. Moreover, our evaluation reveals that the performance of ENTM-TS surpasses or aligns with the optimal outcomes obtained from individual topic models across the two datasets, establishing it as a robust and effective enhancement in text classification tasks.

Keywords Topic model, Topic selection, Feature Selection, Ensemble learning, Text classification, Machine learning

The trend of unstructured digital content published daily via the World Wide Web is increasing, and without any automated aid, it is becoming cumbersome for humans to use it properly. Automatic text classification is a supervised learning task that deals with the problem of assigning predefined categories to textual documents based on their content[1]. Some text classification problems deal with high-dimensional feature space, and determining an appropriate representation of text documents might be crucial for classification performance.

Topic models (TM) are unsupervised machine-learning algorithms for finding latent semantic structures in extensive text documents. They initially emerged as a text-mining technique to discover meaningful hidden patterns and interpretable semantic concepts in a text (topic discovery). However, they have found application in text classification, where the extracted topics from an extensive text collection form the features for document representation[2].

In this regard, TextNetTopics is an innovative topic modeling-based text classification approach. It conducts feature selection by choosing the most highly ranked topics, where a topic represents a group of terms identified by the topic model, to train the classifier. This approach accomplishes dimensionality reduction while retaining richer thematic and semantic information within the text document representations. TextNetTopics, as a feature selection method, can improve performance much more than other equivalent feature selection methods[3].

[1]Computer Engineering Department, Al-Quds University, Jerusalem, Palestine. [2]Computer Science Department, Al-Quds University, Jerusalem, Palestine. [3]Department of Information Systems, Zefat Academic College, Zefat, Israel. ✉email: daniel2vosk@gmail.com; malik.yousef@gmail.com

Moreover, the interpretability of topic features helps the analysts understand and explain the text classification decisions better.

In this article, we present an innovative approach known as ENTM-TS (Ensemble Topic Model for Topic Selection), which is designed to enhance TextNetTopics' capabilities. ENTM-TS innovatively integrates multiple topic models using the Grouping, Scoring, and Modeling approach. This method is specifically crafted to alleviate the performance variability introduced by employing individual topic modeling methods within TextNetTopics, adding stability and offering a more cohesive and robust framework.

Few studies[4–7] have introduced frameworks that combine multiple topic models in an ensemble approach. These studies aim to enhance the quality and interpretability of generated topics by ensuring greater semantic coherence among topic terms. For instance, Belford and Greene[5] introduced a new ensemble topic modeling approach based on the Weighted Term Co-association (WTCA) method. Their approach captures the stability information of topic modeling solutions, particularly those generated using randomly initialized NMF, across multiple runs on the same document corpus by assessing the consistency with which pairs of terms are associated with the same topic. Afterward, they incorporated semantic similarity information from a word embedding model to enhance this co-association information. This augmentation allows for the inclusion of coherence knowledge derived from the embedding. The derived weighted term co-association information is then utilized to create a more interpretable set of ensemble topic descriptors.

Diverging from this trend, our novel approach, ENTM-TS, prioritizes creating topics with greater discriminative power. These topics are intended as input features for training machine learning algorithms specifically designed for text classification tasks.

By leveraging the strengths of each individual model, ENTM-TS aims to overcome the limitations associated with relying on a single-topic modeling method, providing a comprehensive and improved solution for extracting meaningful and discriminatory information from textual data for text classification tasks. In addition, this approach releases the user from the burden of having to select a single method, offering a more user-friendly and adaptive solution. Moreover, ENTM-TS remains versatile across various topic modeling paradigms, as it focuses on the produced topics rather than the model's specific weights or probabilities. This characteristic makes our approach algorithm-agnostic, allowing it to be applied flexibly to different topic modeling techniques.

Additionally, we have performed a thorough comparative study in this article to evaluate TextNetTopics' performance using various topic modeling algorithms. We trained eleven different topic models. We used the extracted topics for each (a topic is a group of words) as input to the G component in the TextNetTopics tool to select the most compelling topic model regarding their predictive behavior for text classification.

We structured the rest of this research using the following sections: Section "Related work" considers an overview of related research work. Section "TextNetTopics" briefly investigates the TextNetTopics tool. Section "Topic model" explores various topic modeling methods. Section "Detailing the ENTM-TS approach: concepts and applications" offers a detailed elaboration of the proposed approach. Section "Experimental work" presents the experimental work. Section "Results and discussion" provides detailed performance results of the TextNetTopics tool with a discussion. Finally, section "Conclusion" concludes our paper and presents our future research work.

Related work

Various studies utilized topic models in text classification. In this aspect, most studies used the topic distribution output of a topic model in document classification as a document representation. They transformed each document into a dense vector of size k (topic embedding), where each cell indicates the proportion of a specific topic in that document.

Luo and Li[8] used Latent Dirichlet Allocation (LDA) topic model to generate a reduced-dimensional representation of topics and fed them as features to a Support Vector Machine (SVM) to classify the data. LDA obtained better results than document frequency (DF) and principle component analysis (PCA).

AL Salemi et al.[9] used LDA to estimate the latent topics in the corpus and use them as features for AdaBoost.MH. However, the study makes some assumptions about selecting the appropriate topics for each category/document and excluding those with low weight. Such an approach significantly improved AdaBoost.MH performance for text categorization and decreased the computational time of its learning rate.

Alhaj et al.[10] proposed a method that enriches text representation by the latent topics extracted using transformer-based topic modeling (BERTopic) within tweets. The topic distribution is concatenated with the original document representation, a word2vec embedding, to produce contextual topic embeddings, which are fed to different classifiers. The experimental results indicate that enriched representation exceeded the baseline models by different rates.

Glazkova[11] compared three common topic modeling techniques, LDA, Gibbs Sampled Dirichlet Multinomial Mixture (GSDMM), and BERTopic, for age-based text classification of Russian books. The study assesses the effect of topic modeling features (document topic distribution vectors) on several machine learning methods, including Logistic Regression (LR), Linear Support Vector Classifier (LSVC), and Multilayer perceptron (MLP). In most cases, topic-enriched classifiers outperform the baselines.

Rijcken et al.[12] compared various topic models in terms of interpretability and predictive performance using the topic embeddings of electronic health records (clinical notes). In this regard, no model exceeds the others on both variables. However, Product-of-Experts LDA (ProdLDA) and Latent Semantic Indexing (LSI) achieve the best predictive performance.

Concerning the use of the topic-word matrix byproduct of a topic model, a minority of studies utilized it as a topic-based document representation.

Zrigui et al.[13] used LDA to represent each document using real-valued features, including words in each topic. Consequently, the approach reduces the VSM vector's dimensionality while preserving semantic and syntactic information in the document representation.

Zhang et al.[14] used LDA with Gibbs Sampling as a feature selection method for text classification. They took only the best terms in each topic by calculating the entropies of the terms on the term-topic matrix and choosing the terms with lower entropy values. Then, use only those features to train a state-of-the-art classifier. According to the experimental results, the proposed approach achieved better classification accuracy while reducing the feature space.

Taşcı et al.[15] conducted a similar study. They extracted the hidden topic utilizing LDA for feature selection using Variational Expectation Maximization for estimation instead of Gibbs sampling. They compared it with the traditional feature selection techniques (i.e., Information Gain, Chi-square, and Document frequency). The results showed that the LDA-based metrics perform similarly to document frequency and chi-square.

Al-Salami et al.[16] utilized Labeled LDA (LLDA) as a feature selection algorithm. LLDA is a supervised version of LDA, which restricts the number of topics to be equivalent to the number of corpus categories. The study employed the LLDA's topic-word distribution matrix, selecting high-weight words for training the classification model. Experimental results demonstrated that LLDA for feature selection accelerated AdaBoost.MH for multi-label categorization and outperformed other methods (i.e., Chi-square, GSSC, and Information Gain).

Mo et al.[17] employed LDA for modeling topic terms and distribution in the corpus and used them as features to train an SVM classifier. Their study compared three feature representation schemes: topic distribution-based features, term enriched-topic features (replacing some LDA single terms with multi-word terms identified using a TerMine, an automatic term recognition tool), and bag-of-words (BOW) features. The outcomes from this research demonstrate that the topic-based features outperform the BOW one when applied to automatic citation screening.

Aguiar et al.[18] used the top-ten most representative words in each topic discovered by BERTopic, along with the topics' distribution to represent each legal document as a feature vector structured by them to classify Brazilian lawsuits. According to the obtained results, it outperformed the baseline.

Considering the studies that use the topic-word matrix, they used all extracted topics as representative features. Even though this approach significantly reduces the document vector dimension, some topics may add noise to the classification model and reduce its performance[3].

TextNetTopics

TextNetTopics is a novel approach developed by Yousef and Voskergian in[3] that performs topic selection instead of the traditional methods of selecting individual words. A topic is a set of words that a specific algorithm might detect from a collection of texts (i.e., a topic model). TextNetTopics acquires its generic approach from the G-S-M (G stands for Grouping, S for Scoring, and M for Modeling) approach[19,20], which was developed by Yousef and his colleagues and utilized primarily on biological data. A review focusing on feature selection methods that group features can be found in this paper[21]. Recently, an enhanced version, TextNetTopics Pro[22], has been developed as a novel text classification framework designed specifically for short text.

Different bioinformatics tools have adapted the Grouping-Scoring-Modeling (G-S-M) approach for integrated biological knowledge through various computation tools such as SVM-RCE[23–25], SVM-RNE[26], maTE[27], CogNet[28], mirCorrNet[29], miRModuleNet[30], integrating Gene Ontology[31], PriPath[32], GediNET[33], miRdisNET[34], GeNetOntology[35], 3Mint[36], and miRGediNET[37]. For more information, refer to a review paper on G-S-M approaches in[19,21].

TextNetTopics tool starts by extracting latent topics, each containing a group of co-occurring words comprising relatively few words (T component). Then, it conducts topic selection rather than word selection by following these steps. First, it creates topic-based sub-datasets, each consisting of words belonging to a topic attached to the original class labels (G component). Second, it sends these sub-datasets to a machine learning algorithm to score and rank topics accordingly (S component). Finally, it uses the top significant topics in an accumulation order to train the classifier and get the performance table (M component). The subset of top-ranked topics with the highest performance will be considered for creating the final model[3].

TextNetTopics uses LDA as a default setting to extract topics. This study investigates the impact on TextNetTopics performance using alternative topic modeling methods, such as Non-negative Matrix Factorization[38], Latent Semantic Indexing[39], Fuzzy Latent Semantic Analysis[40], Probabilistic Latent Semantic Analysis[41], Correlated Topic Model[42], LDA2VEC[43], TOP2VEC[44] BERTopic[45], CombinedTM[46], and Embedded Topic Model[47], respectively, within the T component. For more information about the T component, we refer to[3].

A primary limitation of TextNetTopics, when coupled with a specific topic model, is its inconsistency in performance across various datasets. This inconsistency and perturbation issues require users to experiment with different topic models for a given task to identify the optimal subset of topics from a textual dataset for training the classifier. In other words, the effectiveness of a specific topic model as a feature selection technique dramatically depends on its ability to align with the problem's structure and capture the inherent data patterns. Hence, a certain level of understanding of the internal workings of existing topic modeling algorithms is typically necessary to select a method suitable for the specific problem at hand.

Our research objective is to devise an approach that mitigates the performance variability introduced by employing individual topic modeling methods with TextNetTopics. This approach utilizes the concept of ensembling in feature selection, which aims to capitalize on the strengths of individual feature selectors (i.e., TextNetTopics with various topic modeling methods) while overcoming their weaknesses, ultimately resulting in more stable and reliable topic selection outcomes, which will assist in discriminating the documents of different classes more effectively, and as a consequence, enhancing the performance of a text classification system.

Furthermore, this methodology provides the significant advantage of exempting the user from the responsibility of selecting the most suitable topic modeling technique for a specific context, thereby simplifying the decision-making process in scenarios where multiple methodologies could be applicable.

This study conducts a comparative analysis between the proposed ENTM-TS approach and TextNetTopics, employing various individual Topic Model methods.

Topic model

Topic Modeling, also known as Topic Detection, Topic Extraction, or Topic Analysis, is a statistical text-mining technique with algorithm sets that reveal, uncover, and annotate the underlying thematic and semantic structure from a considerable collection of unlabeled documents. The primary drive of topic analysis is to discover hidden concepts, or latent variables, usually referred to as topics. In addition, TM algorithms provide dimensionality reduction through algebraic and statistical perspectives[48,49].

Topic Modeling has found applications in various fields, such as automatic document indexing, entity relationship discovery, temporal topic trends, and text classification[1].

Regarding the latter task, performing text classification through topic modeling can be achieved by training a topic model that finds two quantities: (1) K latent topics, each associated with a group of words with probabilities (weights). (2) Topic distributions, each value represents a probability of a specific topic given document. Afterward, use those topics as features and train a classifier (e.g., Random Forest, SVM) to calculate the most likely label on new documents[2].

In this area, many well-known topic modeling methods engaged in text analysis in multiple fields are categorized as.

- Algebraic models, including Latent Semantic Analysis/Indexing (LSA/LSI), Non-negative Matrix Factorization (NMF), and Fuzzy Latent Semantic Analysis (FLSA).
- Probabilistic models, including Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), and Correlated Topic Model (CTM).
- Embedding-based models, including LDA2VEC, TOP2VEC, BERTopic, CombinedTM, and Embedded Topic Model (ETM).

Algebraic topic models

(a) *NMF*: Also called PMF - Positive matrix factorization is a geometrically linear algebra optimization approach. NMF-based models learn the hidden thematic information (topics) in the documents by approximately factorizing the high-dimensional term-document matrix V, a bag-of-word matrix representation of a dataset, into two lower-dimensionality matrices $V \approx W * H$, where W represents the term–topic matrix, and H represents the topic–document matrix. The fact that the produced vectors are non-negative contributes to their ease of inspection, especially the interpretability of the topics produced[38].

(b) *LSA*: On the other hand, the latent semantic analysis technique, also called Latent Semantic Indexing, uses mathematical computations on a large text dataset to extract words' contextual meaning and synonyms[50]. LSA utilizes a reduced-rank singular value decomposition (SVD) to reduce the high dimensionality that characterizes the vector space model. It transforms the terms-dimensions of the original term-document matrix (X) into a reduced k-dimensional approximation in the LSA space, preserving only the k most significant singular values and their associated vectors and setting the remainder to zero[51]. Transforming the space into a latent semantic space enables LSA to provide information well beyond the lexical level and reveal the semantical relation between entities of interest. Regardless, LSA suffers from complicated mathematics and has no solid statistical foundation[52].

(c) *FLSA-W*: Similar to LSI, FLSA uses SVD to reduce data dimension. It utilizes the hypothesis that SVD projects words into a lower dimensional space, where semantically related words are located closer to each other. Unlike the original FLSA, which takes the D (Document-concept) matrix, FLSA-W takes the T (Terms-concept) matrix and performs fuzzy c-means clustering to find different topics. In other words, FLSA-W performs word clustering instead of document clustering in FLSA. Finally, it uses linear algebra and Bayes' theorem to find the output metrics (terms-topics and document-topics)[53].

Probabilistic topic models

Probabilistic Models came into play to enhance algebraic approaches by adding probability sense using generative model approaches[54].

(a) *PLSA*: A more principled framework with statistical and probabilistic perspectives is the probabilistic latent semantic analysis (PLSA). PLSA is used to analyze textual data based on a mixture decomposition emanating from the latent variable model, called the aspect model, where the model parameters are estimated via likelihood maximization using the Expectation-Maximization (EM) algorithm. Compared to the standard LSA, Probabilistic-LSA has a sound statistical foundation and provides better performance regarding precision-recall metrics on a set of small document collections[52].

(b) *LDA*: One widely utilized and highly studied topic modeling method in the literature is the latent Dirichlet al.location. It is a fully generative, thematic probabilistic topic model. LDA is an unsupervised method for discovering topics based on a latent topic model. The leading idea behind LDA is that each document in a dataset is modeled as a sparse mixed-membership of topics, where a topic represents a multinomial distribution over a fixed vocabulary that defines the likelihood of each word appearing in a given topic. Following this approach, these topics' distribution will concisely express each document in the dataset. LDA

is a generalization of PLSA by adding a Dirichlet prior distribution over topic-word and document-topic distributions and uses Bayes estimation instead of maximum likelihood estimation[48,55–57].

(c) *CTM*: A limitation of LDA is the inability to model the correlation between discovered topics, which stems from the independence assumptions implied in the Dirichlet distribution on the topic proportions. A correlated topic model (CTM) builds on the LDA model and exhibits correlation by replacing Dirichlet with the logistic normal distribution, incorporating a covariance structure among the variables to capture correlations between topics[42].

Until the time of conducting this study, most studies in LSA, NMF, FLSA, PLSA, CTM, and LDA approaches still use BOW (bag-of-words) document representations as input. The corpus is converted into a term-document matrix, disregarding syntactic and semantic relationships among terms, not accounting for the context of terms in a sentence, and neglecting their order, retaining only the terms count in the document. Such representation may fail to represent documents accurately. Few researchers consider the sequence of words (Bi-gram and N-gram) during topic modeling[49].

Embedding-based topic models

Recently, many algorithms have emerged to improve the quality of topic modeling by taking advantage of recent advances in distributed dense word vectors, which capture the semantic and syntactic regularities in language and the sentential coherence to build document-level abstractions. These vectors encode the meaning of texts such that similar texts are close in vector space.

(a) *LDA2VEC*: A hybrid model that combines word embedding representation using word2vec's skip-gram architecture with LDA-optimized sparse topic mixtures. This algorithm simultaneously learns dense word, topic, and sparse document-level vectors. It utilizes a modified skip-gram approach that retains local and global information by merging pivot word and document vectors to predict context words more effectively. The unique aspect of LDA2VEC is that each topic has a learned distributed representation within the same vector space as word vectors, facilitating the identification of relevant word vectors associated with topic vectors. Additionally, LDA2VEC enforces sparsity in final document weight vectors, ensuring non-negative values that sum to unity. This allows topic membership to be viewed as percentages rather than unbounded weights, making LDA2VEC capable of creating human-interpretable LDA topic and document representations[43].

(b) *TOP2VEC*: Previous algorithms often require known topic numbers as a model hyperparameter, posing challenges for unfamiliar datasets. TOP2VEC overcomes this by automatically detecting topics presented in documents. The algorithm's fundamental principle assumes that many semantically similar documents indicate an underlying topic; thus, the number of dense areas of documents in the semantic space is assumed to be the number of prominent topics. The mechanism of TOP2VEC starts by jointly generating document and word vectors embedded in the same semantic space using various methods, such as Doc2Vec. TOP2Vec then performs dimensionality reduction by applying UMAP and the HDBSCAN algorithm for identifying variable-dense document clusters. Finally, it uses each dense area to create a topic vector (the document vectors' centroid in the original dimension), and the closest word vectors in order of proximity become the topic words that best describe it semantically. One characteristic feature of TOP2VEC is the ability to perform topic reduction by hierarchically grouping similar topics[44,58].

(c) *BERTopic*: BERTopic, like TOP2VEC, automatically discovers the number of topics. However, a centroid-based perspective in TOP2VEC may misrepresent a topic since a cluster may only sometimes lie within a symmetric sphere around a cluster's centroid. BERTopic overcomes this by leveraging the clustering embeddings approach and a custom class-based variation of TF_IDF to extract topic representations. BERTopic uses the sentence-BERT framework to convert documents to embedding vectors, then employs UMAP to reduce their dimensionality to optimize the clustering process of the HDBSCAN algorithm. Finally, it treats all documents within a cluster as a single document and adjusts TF_IDF to account for this representation. The aim is to measure the importance of a term to a cluster (topic) instead of an individual document. Like TOP2VEC, BERTopic allows the number of topics to be reduced to a user-specified value by merging the least common topic with its most similar one[45].

(d) *ETM*: The embedded topic model addresses the problem of traditional topic modeling accommodating large and heavy-tailed vocabularies (including stop and rare words). Similar to LDA, ETM is a generative probabilistic model. However, words and topics are represented by embedding representations. In addition, the distribution of terms within a topic is equivalent to the exponentiated inner product of the topic's embedding and each term's embedding. In other words, it assigns a probability to a word in a specific topic by measuring the agreement between word and topic embedding[47].

(e) *CombinedTM*: The Combined Topic Model aims to enhance topic coherence and quality by extending Product-of-Experts LDA (ProdLDA), a neural topic model based on the Variational AutoEncoder, to incorporate SBERT contextualized sentence embeddings. It concatenates document and BOW representation and projects it through a hidden layer that directly maps into continuous latent K representation. Finally, a decoder network uses this representation to reconstruct the BOW and generates topics' words[46].

Detailing the ENTM-TS approach: concepts and applications

ENTM-TS

The Ensemble Topic Model for Topic Selection (ENTM_TS) represents a significant development beyond TextNetTopics, offering a novel topic model-based methodology for text classification. This approach integrates various topic models, utilizing ensemble learning principles and methods. ENTM_TS effectively mitigates the

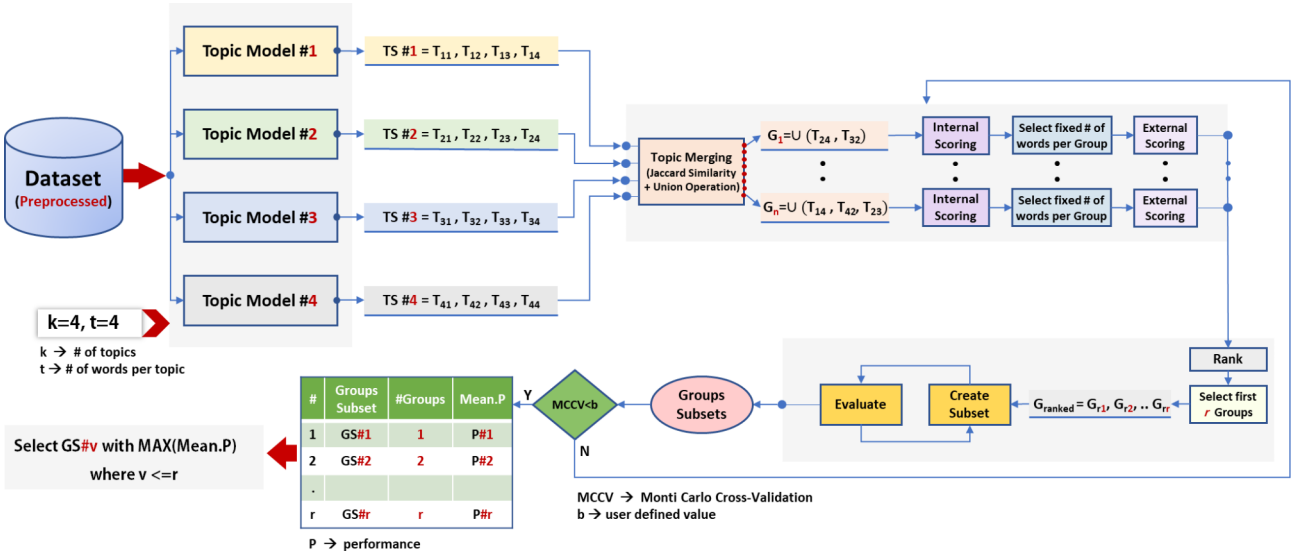


Fig. 1. General framework of ENTM_TS.

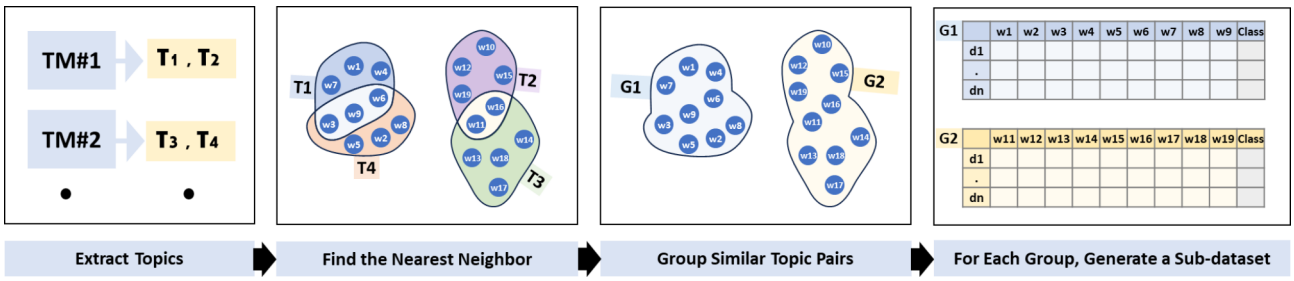


Fig. 2. Visual representation of the grouping component.

performance inconsistencies inherent in individual topic modeling techniques found in TextNetTopics, thereby achieving greater stability and a more unified framework.

In simpler terms, ENTM_TS focuses on identifying the top-ranked *v* groups, each composed of semantically related words aggregated from highly similar topics (extracted by various topic models), to enhance the classification performance. The primary aim of ENTM_TS is to reduce the dimensionality of the vector space model while preserving semantic structures in text and making the text more related and topic-oriented for classification purposes.

ENTM_TS follows the G-S-M (Grouping, Scoring, and Modeling) approach, which consists of three main components:

- Grouping Component: ensures efficient topic extraction via various topic modeling algorithms, minimizes redundancy to guarantee topic diversity through advanced topic merging techniques, and facilitates the creation of groups for subsequent analysis, where a group consists of aggregated words from highly similar topics.
- Scoring Component: utilizes a dual-stage scoring process to comprehensively evaluate group performance, considering internal characteristics and external classification effectiveness. It creates a refined set of groups with enhanced discriminatory capabilities.
- Modeling Component: selects the top-ranked *v* groups to train the final model for optimal performance.

Figure 1 provides the general framework for ENTM_TS.

This section explains our innovative methodology in detail:

Let C denote the textual dataset. We assume the presence of *d* documents and *w* distinct words in our dataset C. The dataset C is divided into C_{train} and C_{test}. C_{train} is employed to score the created groups of topics and train the classifier, forming the model. Conversely, C_{test} is primarily used to test and report the final performance.

Grouping component

This phase encompasses three key steps (refer to Fig. 2):

1. *Topic Extraction*: This step employs j different topic models to extract topics from the preprocessed dataset. Each topic consists of a list of semantically related words. In this aspect, the user has to provide two input

values: *k*, which represents the number of topics to be extracted by a topic model, and *t*, which refers to the number of words per topic. This step yields an initial list of *kxj* topics. However, It is important to note that different topic models might generate similar topics (share common terms), resulting in redundant topics. This similarity between topics can diminish feature selection efficiency, particularly during topic subsets' generation (refer to M component). The generated subsets might exhibit only marginal variations when the top r topics are substantially similar.

2. *Redundancy Minimization via Topic Merging*: To address redundancy, we combine comparable topics. Each topic is encoded as a bit vector, assigning each unique word in all topics a position. The presence of a word in a topic is denoted by one, otherwise zero. We calculate the similarity score between each topic pair using the Tanimoto (Jaccard Index) method, expressed as:

$$J(A, B) = |A \cup B| / |A \cap B|$$

Where A and B are sets representing two topics, $|A \cap B|$ denotes the number of common words between the topics, and $|A \cup B|$ represents the number of all unique words from both topics. The coefficient ranges from 0 to 1, where 0 indicates no similarity, and 1 indicates complete similarity.

For each topic in the initial list generated in the previous step, we determine its nearest neighbor (the topic with which it shares the highest similarity score). Topics that exhibit no similarity or share similarity below a predefined threshold with others remain separate. After obtaining the final pairs of topics (each topic with its closest match), we organize them in descending order based on their scores. The merging process is carried out sequentially by performing a union operation, ensuring that if a topic is involved in two or more pairs, it is only merged with the pair having the highest score. This process will result in a new list comprising merged and non-merged topics, which we will refer to as "groups" for simplicity. This process can be iterated *f* times to progressively diminish redundancy and ensure the diversity among the groups obtained.

3. *Sub-datasets Creation*: The list of groups produced in the previous step and the training Bag-of-Words (BOW) dataset are utilized to create group-based sub-datasets. Each sub-dataset corresponds to a specific group and comprises a Bag-of-Words (BOW) representation containing solely the words associated with that group, along with the respective class labels (positive or negative) of the training documents.

Scoring Component: This component aims to normalize groups size and extracts the highly discriminative groups for training the classification model. The scoring process unfolds in two stages: internally and externally (refer to Fig. 3).

1. *Internal Scoring*: In the preceding stage, groups were generated with varying numbers of words. This variability introduces a potential bias, as groups with larger sizes may exhibit seemingly superior performance. To address this, we propose a refinement: normalizing for group size by extracting a fixed number of words from each group. This approach ensures a fair comparison, eliminating the influence of group size on performance metrics and providing a more accurate assessment of the inherent quality and discriminative power of each group.

   Firstly, individual subdatasets, each representing a distinct group, undergo evaluation through the XGBOOST algorithm. This step determines the importance values assigned to each word within the given group. Post-analysis, a predefined number of words exhibiting notably high importance are identified and selected from each group. This strategic selection of crucial terms within each group enhances the precision of subsequent phases in the analysis. The output of this stage is refined group-based subdatasets.

2. *External Scoring*: With the refined group-based subdatasets in place, a stratified k-fold cross-validation technique, incorporated with a machine learning model, such as Random Forest, is applied to each subdataset to assign a score to each group (e.g., mean F1-score). This score assesses the performance of the generated model, which serves as a metric indicating the group's effectiveness in distinguishing between classes within

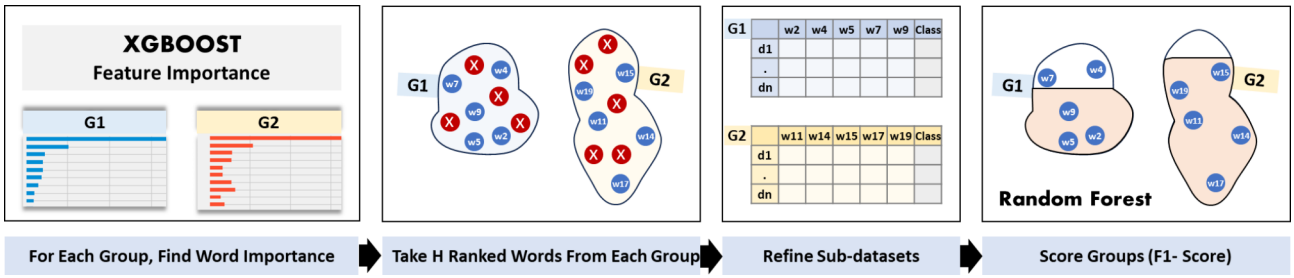


Fig. 3. Visual representation of the scoring component.

a text classification problem. Following the evaluation of each group, they are ranked based on their respective mean F1-scores. The highly ranked *r* groups, indicative of superior discriminatory capabilities and stronger classification potential, are prioritized for further investigation.

Modeling component:

During this phase, top *r* ranked groups are sequentially accumulated in a linear manner. The process begins with the top-ranked group, and in each iteration, an additional group is incorporated until the top *r* ranked groups are included. In each iteration, the accumulated groups collectively define a set of aggregated words. These aggregated words are utilized to construct training and testing subdatasets, drawing from the training and testing Bag-of-Words (BOW) dataset, respectively. Subsequently, these subdatasets are employed to train and test the model.

This iterative procedure is repeated for all possible combinations of accumulated groups. Following the model evaluation for each combination, the subdataset with the highest performance, as determined by the specified metric (F1-score), is chosen for training the final model (refer to Fig. 4). In other words, the M component selects the top *v*-ranked groups, where *v* is less than or equal to *r*, that provides the best performance (i.e., best discriminative power) for the specified classification task.

Time complexity for ENTM-TS

Since ENTM-TS and TextNetTopics share the Modeling component, this section compares the time complexity between their Grouping and Scoring components.

The time complexity of the ENTM-TS Grouping component can be analyzed as follows:

- *Topic Extraction*: The time complexity of this step involves employing *j* different topic models to extract topics from a preprocessed dataset. It can be summarized as $O(j \cdot d \cdot w + t \cdot k \cdot j)$, where *d* represents the number of documents in the dataset, *w* is the average number of words per document, *k* is the number of topics to be extracted by each topic model, and *t* is the number of words per topic. The algorithm iterates over all documents for each topic model, contributing $O(d \cdot w)$ operations per topic model. Additionally, selecting *t* words for each of the *k* topics in each topic model adds *t·k·j* operations.
- *Redundancy Minimization via Topic Merging*: Calculating the similarity score between each pair of topics involves comparing every topic with every other topic. For each comparison, we need to compare two sets of words, each having *t* words, resulting in $O(n(n-1) \cdot t^2)$ operations, where *n* is the total number of topics extracted by different topic models. Organizing the topics based on their scores, selecting the nearest topic neighbor, and performing the merging operation can be done in $O(\mathrm{n} \cdot \mathrm{logn})$ time complexity if an efficient sorting algorithm is used.
- *Sub-datasets Creation*: Creating group-based sub-datasets involves iterating over the training Bag-of-Words dataset for each group. If the average number of words associated with each group is denoted by w, and the total number of groups is denoted by g, then the time complexity for this step is $O(g \cdot w)$.

Therefore, the overall time complexity of the Grouping Component is $O(j \cdot d \cdot w + t \cdot k \cdot j + n^2 \cdot t^2 + n \cdot logn + g \cdot w)$. The dominant factor in this time complexity is likely to be the Topic Extraction step if the number of topic models and topics per model is large. The Redundancy Minimization via the Topic Merging step could also be significant, especially if *n* (the total number of topics) is large. In contrast to ENTM-TS, TextNetTopics Grouping component only involves employing one topic model (.e., LDA) to extract topics from a preprocessed dataset and involves sub-dataset creation. It is time complexity can be summarized as $O(d \cdot w + t \cdot k + g \cdot w)$.

The time complexity of the ENTM-TS Scoring component can be analyzed as follows:

- *Internal Scoring*: For each group, the XGBOOST algorithm evaluates the importance values of each word. Let's denote the average number of words per group as *w*. The time complexity for this step is $O(g \cdot w)$, where g is the total number of groups. The selection of words with high importance values involves sorting the words by importance and selecting the top *h* words. Sorting would require O(*w·log(w)*) operations, which, for a fixed w, can be considered as O(1) in the context of the entire operation since w is typically small (e.g., 20 or 30 words).

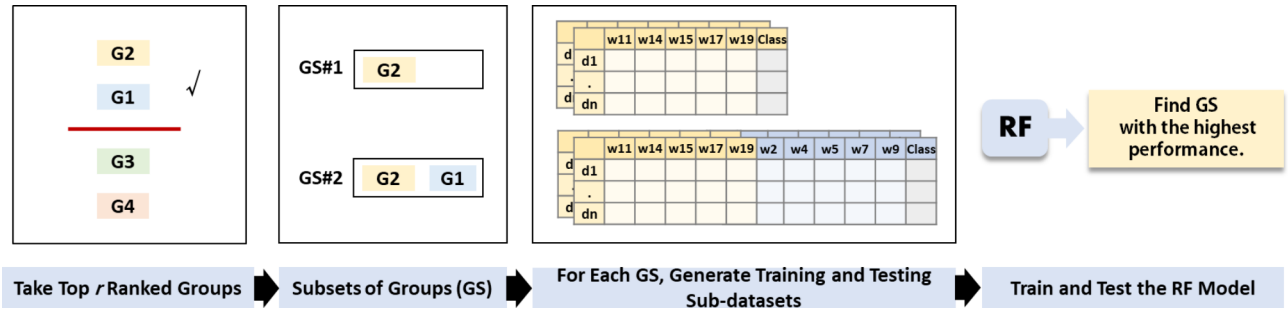


Fig. 4. Visual Representation of the Modeling Component.

This operation is performed for each group, resulting in $O(g \cdot 1) = O(g)$ complexity. Therefore, the overall time complexity for internal scoring is $O(g \cdot w)$.

- *External Scoring*: In this stage, the Random Forest model is trained and evaluated for each group using Monte Carlo cross-validation, with z iterations. This process involves training the model z times (once for each fold) for each of the g groups, resulting in a time complexity of $O(g \cdot z \cdot q.h \cdot log(h))$, where q is the number of trees in the forest, h is the number of words per group, and $log(h)$ accounts for the sorting of features at each split. This is a simplified estimate and may vary depending on the specific implementation details and optimizations used in the Random Forest training algorithm. Subsequently, ranking the groups based on their mean F1 scores requires sorting them by score, which can be done in $O(g \cdot log(g))$ operations. Combining both stages results in a total time complexity of $O(g \cdot (w + z \cdot q.h \cdot log(h) + log(g)))$. For a fixed g, this time complexity can be considered as $O(g \cdot z \cdot q.h \cdot log(h))$ in the context of the entire operation since g is typically small (e.g., 40 groups). On the other hands, TextNetTopics Scoring component only involves external scoring for k topics (each consisting of t terms), where $k < g$, resulting in a time complexity of $O(k \cdot z \cdot q.t \cdot log(t))$.

While our proposed ensemble approach may operate slower due to its topic generation, integration, and internal scoring steps, it is essential to note that ENTM-TS primary objective is to generate highly discriminative topics. This tradeoff between computational expense and improved performance is a key consideration.

Experimental work

The subsequent subsections briefly explain the dataset used in this study, the NLP techniques employed in processing the dataset, and finally, we describe the experimental setup.

Dataset

In this study, we empirically evaluated our experiments using the Drug-Induced Liver Injury (DILI) training dataset from the CAMDA community (international conference on critical assessment of massive data analysis) and a Web of Science (WOS) document classification dataset[59].

CAMDA collected a large set of PubMed articles relevant to DILI, representing the positive class, with an additional set of irrelevant articles forming the negative class. The training dataset comprises 14,000 titles and abstract sections divided equally between both classes.

We also used the WOS − 5736 dataset, containing three higher-level classes. Two have four sub-classes (categories), and the last has three subclasses (categories). To evaluate TextNetTopics, we converted the dataset into two balanced classes. We chose the one with the highest number of documents (# 2847 abstracts) as a positive class and merged the remaining two (#1597 and #1292 abstracts) to create a negative class.

Data preprocessing

The raw form of documents in the dataset may contain redundant, irrelevant, noninformative, and noisy data. Their presence may increase the computation time and necessitate more memory to run machine learning algorithms; they may also degrade the classification performance. Thus, a text-preprocessing phase is crucial to get the best of the analysis and reduce the input data's dimensionality.

We employed Knime workflows for several NLP tasks, including filtering out punctuation, non-English text, stop words, terms primarily composed of digits, and terms with fewer than three characters. Additionally, we applied a minimum document frequency threshold of 1% to exclude less frequent terms, performed case-folding to standardize text cases, and conducted stemming using the snowball stemming library to reduce words to their root forms.

After performing the preprocessing phase for the DILI-CAMDA and WOS − 5736 datasets, the terms were reduced to 1167 and 1298, respectively.

To input the dataset to the TextNetTopics tool, we need to convert each unstructured free text document into a structured numerical form called a feature vector. In this project, we constructed a bag-of-words feature vector representation for each document in the corpus. We have used the relative TF format as a term-weighting method, where each value in a document vector is computed by dividing the count of the respective term by the total number of terms in a document.

Experimental setup

We performed all the NLP preprocessing tasks using the Knime workflow that one can download from Knime Hub[60]. To evaluate TextNetTopics and ENTM-TS tools, we used the Knime implementation workflows, also found in the Knime hub[60]. To evaluate the topic models used in this study, we implemented them in one Google collab notebook accessible on GitHub[61] utilizing various Python-based libraries, except LDA, where we used its Knime implementation.

To assess the probabilistic models, we used the PLSA library[62], a Python implementation of Probabilistic Latent Semantic Analysis. We used tomotopy[63], a Python extension to tomoto (topic modeling tool), which provided us with the CTM algorithm. We used the LDA implementation in Knime, a simple parallel threaded implementation of LDA, following the 'Distributed Algorithms for Topic Models' introduced by Newman, Asuncion, Smyth, and Welling[64]. We set the alpha parameter (Dirichlet prior on the per-document topic distributions) representing a document-topic density and the beta parameter (Dirichlet prior on the per-topic word distribution) representing a topic-word density to their default value, 0.1 for each. We ran LDA for 1000 iterations to estimate the topics.

To assess the algebraic models, we utilized the scikit-learn library[65] to extract topics from LSA and NMF topic modeling methods. For FLSA-W, we used the FuzzyTM python package[66,67] and set the following parameters (cluster method: fuzzy-c means clustering, SVD factors = 2, word_weigting = normal).

Concerning the embedding-based topic models, for instance, the LDA2VEC topic model, we have used the Tensorflow implementation of Chris Moody's LDA2VEC, publicly available at[68]. To extract topics from LDA2VEC, we have trained the model for 400 epochs with a batch size of 4096. We initialized the embedding layer with a pre-trained glove word embedding model, which provides word embedding vectors of 200 dimensions. Regarding the TOP2VEC algorithm, this research utilized the TOP2VEC library[58] in Python, with the following hyperparameters: setting the speed to' deep-learn' for learning the best quality vectors and using Doc2Vec model to generate the joint word and document embeddings.

With respect to a BERTopic-based approach, we used the BERTopic Python library available at[69] with the default configurations, except the min_topic_size was set to 25. For combinedTM, we used the contextualized_topic_models Python library[70]. To extract topics, we used two files: a preprocessed text for the bag of word creation and a not-preprocessed text for BERT embedding creation. However, this approach has one restriction: The rest of the document will be lost if the document size is longer than 128 tokens (SBERT's sentence-length limit). Furthermore, we used the ETM[71], which simultaneously finds an embedding space and discovers topics as part of the fitting process. We used the word2vec embedding model and set the training epochs to 300.

This research work considers two main parameters shared with all the previous algorithms. The number of topics and the number of words per topic are set to twenty since they result in a better performance than other values. Although TOP2VEC and BERTopic automatically produce the number of topics, i.e., in the CAMDA dataset, TOP2VEC generated 39 topics, and BERTopic generated 61 topics, we used the topic reduction feature to merge semantically similar small topics into 20 most representative topics. A similar k-value hyper-parameter configuration enabled us to compare fairly with other utilized methods.

For ENTM-TS, we executed topic merging three times to robustly minimize redundancy. In the internal scoring phase, we normalized the group size by extracting ten words from each group, ensuring a balanced and informative representation. Subsequently, in the external scoring phase, we deliberately selected the top ten ranked topics to serve as inputs for the M component.

Finally, we utilized a stratified Monte Carlo Cross-Validation (MCCV) to evaluate the TextNetTopics and ENTM-TS performances, repeated ten times. Each time, the dataset was split randomly into 90%-10%; 90% of the documents were retained for training, and the remaining were used for testing.

Evaluation

The evaluation measures utilized in this study are the standard performance measures: accuracy, recall, precision, f1-score, area under the ROC curve (AUC), specificity, and Cohen Capa. The results section will consider the F1-score as the primary performance evaluation and analysis measure.

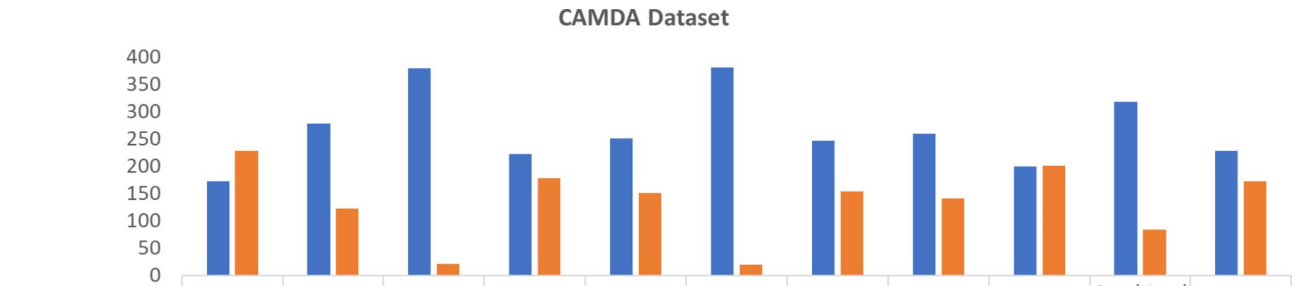
Results and discussion

Accumulated topics distinct terms

When TextNetTopics accumulates topics to train the model, different topics may share common terms, creating duplicate terms (redundant features). According to the literature, the presence of duplicate words in topic modeling outputs can vary depending on the specific method and dataset used. While some methods may produce more duplicate words than others, the number of word duplications can depend on several factors, including the dataset's nature, the topics' complexity, and the evaluation criteria used. Figures 5 and 6 show the percentage of distinct and duplicate terms when we accumulate the first 20 topics in both datasets, where each discovered topic contains 20 terms.

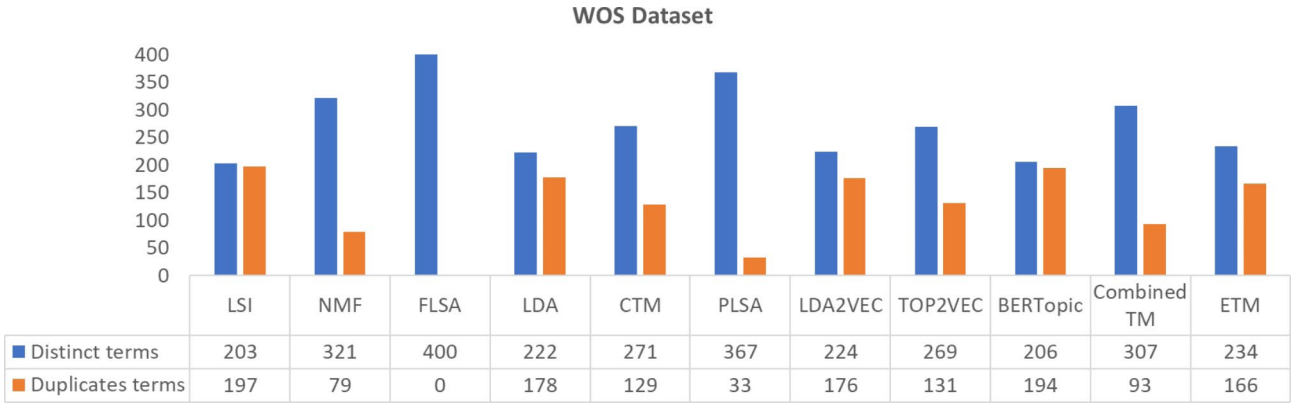
In the CAMDA dataset, for a total of 400 terms, PLSA and FLSA have the lowest overlapping terms, with 5% only, then CombinedTM, NMF, TOP2VEC, CTM, and LDA2VEC, with 21%, 31%, 35%, 38%, and 39%, respectively, while ETM, LDA, BERTopic, and LSI have 43%, 45%, 50% and 57% of duplicate terms. Sharing the exact words between topics may reduce their interpretability. According to[72], FLSA handles such redundancy.

In the WOS−5736 dataset, we got similar results. FLSA has no shared terms. PLSA has the lowest overlapping terms, with only 8%, followed by NMF and CombinedTM, with 20% and 23%. CTM, TOP2VEC,



	LSI	NMF	FLSA	LDA	CTM	PLSA	LDA2VEC	TOP2VEC	BERTopic	Combined TM	ETM
Distinct terms	172	278	379	222	250	380	246	259	199	317	228
Duplicate terms	228	122	21	178	150	20	154	141	201	83	172

Fig. 5. CAMDA dataset—the count for distinct and duplicate terms in 20 topics, where each uncovered topic contains 20 terms.



	LSI	NMF	FLSA	LDA	CTM	PLSA	LDA2VEC	TOP2VEC	BERTopic	Combined TM	ETM
Distinct terms	203	321	400	222	271	367	224	269	206	307	234
Duplicates terms	197	79	0	178	129	33	176	131	194	93	166

Fig. 6. WOS dataset—the count for distinct and duplicate terms in 20 topics, where each uncovered topic contains 20 terms.

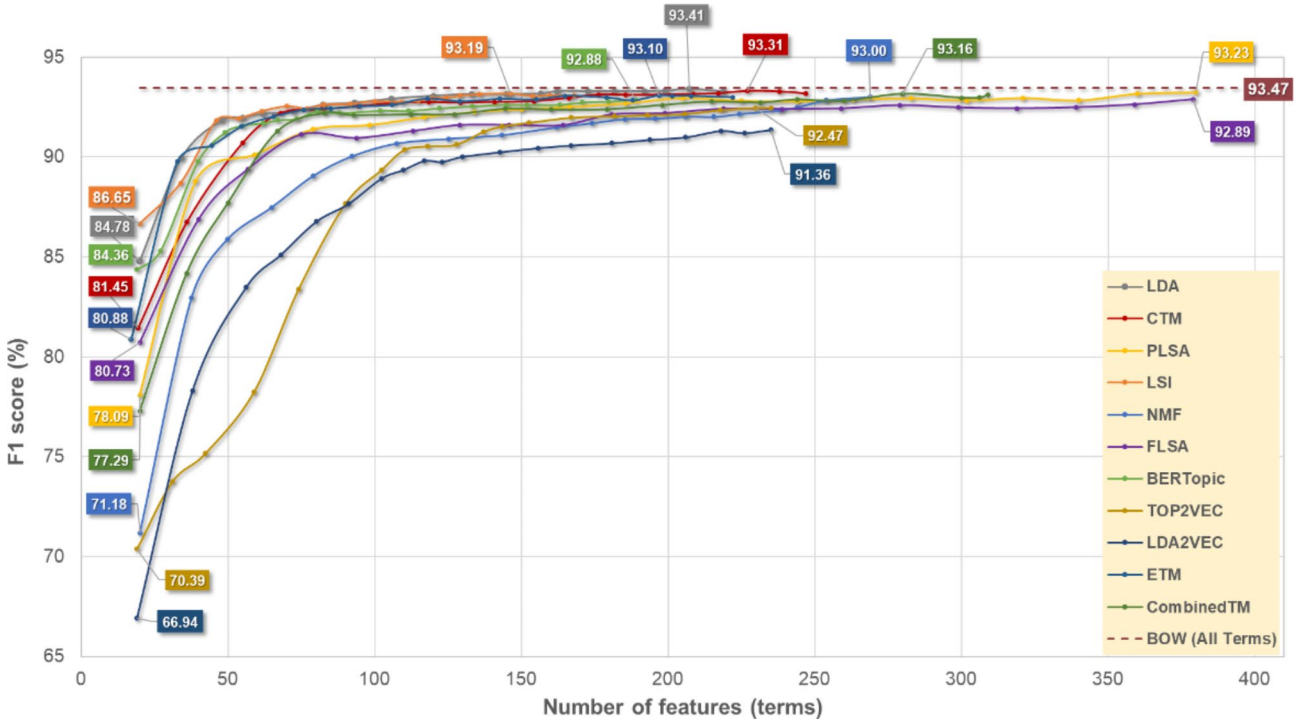


Fig. 7. TextNetTopics performance results over accumulated top topics (20 topics) for the CAMDA dataset utilizing different topic modeling methods in the T component. Symbols on the line represent the number of accumulated topics.

ETM, LDA2VEC, and LDA have 32%, 33%, 42, 44%, and 45% of shared terms, respectively, while BERTopic and LSI both have 49% of duplicate terms.

Evaluating TextNetTopics performance using various topic modeling methods in the T component

Figures 7 and 8 present TextNetTopics performance results over different constructed feature sets (i.e., the top one ranked topic, top two ranked topics, until top twenty ranked topics) extracted from NMF, LSI, FLSA, PLSA, LDA, CTM, LDA2VEC, TOP2VEC, BERTopic, CombinedTM, and ETM, of the DILI-CAMDA training dataset. As we increase the feature set (accumulated top-ranked topics in our case), we gain an increase in all the performance measures (i.e., Accuracy, Precession, Recall, F1-score, and AUC). Such results indicate the significance of discovered topics in discriminating the classes in this classification problem. In some topic modeling methods, adding additional topics may add noise to the feature set and reduce the performance of the resulting model; in such a case, TextNetTopics can discover the correct number of topics to accumulate in order to produce an optimal set of features to proceed toward a lower time cost and yield the best performance.

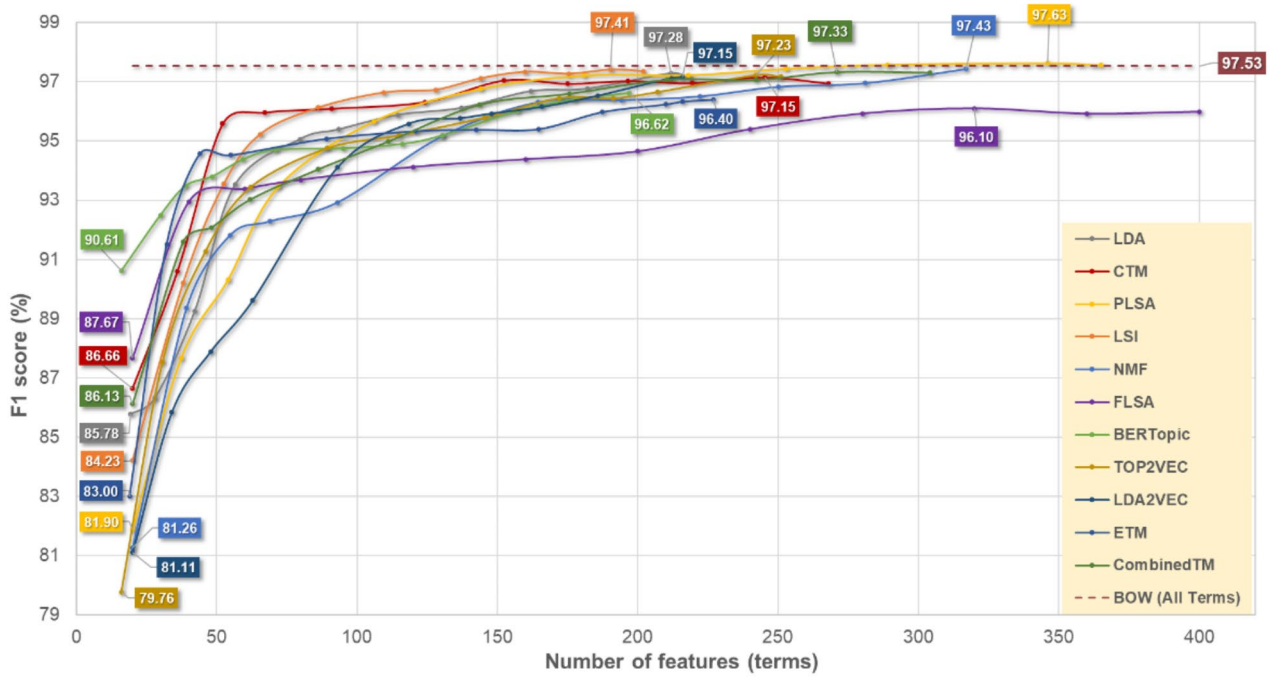


Fig. 8. TextNetTopics performance results over accumulated top topics (20 topics) for the WOS dataset utilizing different topic modeling methods in the T component. Symbols on the line represent the number of accumulated topics.

Finally, providing high-performance measures at lower feature sets is clear evidence that TextNetTopics can fulfill the dimensionality reduction while preserving highly representative and discriminative features in the text document representations.

According to Fig. 7, Probabilistic models achieved the highest F1 score. For instance, LDA got 93.41% over the top 19 topics with 207 features/terms, CTM got 93.31% over the top 19 topics with 227 features/terms, and PLSA got 93.23% over the top 20 topics with 380 features/terms. Moreover, Algebraic and some Embedding-based topic models yielded similar results. For instance, LSI got 93.19% over the top 17 topics with 146 features/terms, CombinedTM achieved 93.16% over the top 17 topics with 280 features/terms, ETM acquired 93.16% over the top 17 topics with 197 features/terms, NMF yielded 93.00% over the top 20 topics with 269 features/terms. FLSA and BERTopics obtained 92.89% and 92.88%, respectively, over the top 20 topics with 379 and 189 features/terms. Regarding the remaining Embedding-based models, although they resulted in comparable results, their F1 score was the lowest. For example, TOP2VEC and LDA2VEC got 92.47% and 91.36% over the top 20 topics with 235 features/terms.

Another interesting fact is that the majority of topic models reach the performance of using all terms at 1167 (preprocessed dictionary). Using LDA, for instance, we can obtain an 82% reduction in the corpus size without any loss in performance.

Although no model outperforms other models (makes a statistically significant difference) in terms of the F1-score, regarding the number of features used, LSI and LDA achieve high-performance results at a lower feature set. For instance, using the top-one ranked group only (20 terms) to train the classifier, LSI achieves 86.65%, and LDA yields 84.78%. Using the accumulated top-ten ranked groups (~100 terms) to train the classifier, LSI achieves 92.80%, and LDA yields 92.91%. In higher accumulated groups, both have similar performance. Moreover, to provide the same F1-score of 93.00%, LSI and LDA use approximately 66%, 60%, 55%, 45% and 26% fewer features (terms) than PLSA, CombinedTM, NMF, ETM, and CTM, respectively.

Figure 8 presents TextNetTopics performance results over accumulated top topics for the WOS−5736 dataset using different topic modeling methods in the T component. According to Fig. 8, PLSA achieved the highest F1 score with 97.63% over the top 19 topics with 346 features/terms, then NMF got 97.43% over the top 20 topics with 317 features/terms. LSI yielded a similar result to NMF; it gained 97.41% over the top 19 topics with 190 features/terms, then CombinedTM acquired 97.33% over the top 19 topics with 271 features/terms. LDA obtained 97.28% over the top 19 topics with 212 features/terms. TOP2VEC got 97.23% over the top 19 topics with 242 features/terms, then LDA2VEC achieved 97.15% over the top 20 topics with 216 features/terms. CTM got the same result as LDA2VEC with 97.15% over the top 19 topics with 245 features/terms. BERTopic got 96.62% over the top 20 topics with 197 features/terms. Interestingly, ETM and FLSA got the lowest F1 score. ETM got 96.40% over the top 20 topics with 227 features/terms, and FLSA got 96.10% over the top 18 topics with 320 features/terms. Concerning the number of features used, LSI in this dataset also achieves high-performance results at a lower feature set.

Topic Model	Accuracy	Recall	Precision	F1-score	AUC	Specificity
LDA	**93.18**	**92.85**	93.52	**93.18**	97.91	93.50
LSI	93.09	92.22	**93.92**	93.06	**97.99**	**93.97**
ETM	92.93	92.27	93.56	92.91	97.80	93.59
CTM	92.79	92.05	93.53	92.76	97.84	93.49
BERTopic	92.62	92.08	93.16	92.61	97.68	93.18
CombinedTM	92.41	91.99	92.84	92.41	97.60	92.83
PLSA	92.41	91.79	93.02	92.39	97.47	93.03
TOP2VEC	91.77	90.82	92.66	91.73	96.59	92.74
FLSA	91.60	91.31	91.88	91.60	97.20	91.92
NMF	91.55	90.41	92.60	91.49	97.01	92.71
LDA2VEC	90.48	89.77	89.77	90.45	96.02	91.20

Table 1. TextNetTopics performance results for the CAMDA dataset, utilizing various topic modeling methods in the T component (150 features). The highest values for each performance metric are highlighted in bold.

	NMF	LSI	FLSA	PLSA	LDA	CTM	LDA2VEC	TOP2VEC	BERTopic	CombinedTM	ETM
NMF	**278**	56%	50%	57%	55%	62%	37%	43%	49%	57%	56%
LSI	90%	**172**	62%	65%	70%	62%	44%	50%	62%	67%	70%
FLSA	37%	28%	**379**	46%	31%	39%	28%	29%	28%	39%	33%
PLSA	42%	29%	60%	**380**	39%	39%	34%	36%	31%	43%	37%
LDA	69%	54%	52%	68%	**222**	58%	41%	45%	55%	63%	64%
CTM	69%	43%	58%	59%	52%	**250**	34%	40%	42%	59%	65%
LDA2VEC	42%	30%	43%	53%	37%	35%	**246**	52%	37%	47%	32%
TOP2VEC	46%	33%	43%	53%	38%	38%	50%	**259**	42%	51%	41%
BERTopic	69%	54%	53%	58%	62%	52%	45%	54%	**199**	63%	57%
CombinedTM	50%	37%	47%	51%	44%	46%	36%	42%	39%	**317**	41%
ETM	68%	53%	55%	62%	62%	71%	35%	50%	50%	57%	**228**

Table 2. The percentage of intersected terms over 20 extracted topics between various topic modeling methods in the CAMDA dataset. The diagonal values represent the number of distinct terms in 20 topics extracted by each topic modeling method.

Again, the majority of topic models reach the performance of using all terms at 1298 (preprocessed dictionary). Using LSI, for instance, we can obtain an 85% reduction in the corpus size without any performance loss.

Observing the obtained results for LSI in both datasets provides clear evidence that it can achieve higher or comparable performance to other topic modeling methods with a reduced number of discriminative features in the text document representations.

Table 1 compares the TextNetTopics performance when utilizing different topic modeling methods in the T component for the DILI-CAMDA dataset. We considered the first 150 features (terms) for comparison. We sorted the table in ascending order according to the F1 score.

We notice that LDA and LSI perform better than other topic models in all the standard performance measures. We notice comparable results between PLSA and BERTopic and between NMF and TOP2VEC. We can attribute the similarities between results to the number of terms' intersections between them. The topics extracted from BERTopic show a 58% term overlap with PLSA across 20 topics, while TOP2VEC shares 46% of its terms with NMF within these 20 topics. It is worth mentioning that LSI exhibits an even more substantial 70% term overlap with LDA. Table 2 provides the number of intersected terms over 20 topics between the topics extracted from the mentioned methods.

Table 3 provides the same comparison, however, for the WOS dataset. We notice this time that LSI and CTM perform better than other Topic models in all the standard performance measures.

Both Tables 1 and 3 offer compelling evidence of the performance variability introduced by different topic models on the same dataset, as well as the performance variability introduced by a specific topic model across various datasets. Therefore, there is a need for a novel approach that utilizes topic modeling algorithms for text classification in a stable way, and offers a more cohesive and robust framework.

Performance evaluation of ENTM-TS and TextNetTopics

In this section, we evaluate our proposed approach (ENTM-TS) by ensembling three algebraic topic models, three probabilistic topic models, and three embedding-based topic models. We compare it with TextNetTopics (TNT), which incorporates a single topic model in each iteration (e.g., PLSA, LDA, etc.).

Topic Model	Accuracy	Recall	Precision	F1-score	AUC	Specificity
LSI	**97.11**	**98.00**	96.26	**97.12**	**99.49**	96.23
CTM	97.06	97.37	**96.75**	97.04	99.35	**96.73**
PLSA	96.76	97.51	96.04	96.76	99.02	96.02
CombinedTM	96.60	97.05	96.15	96.60	99.34	96.16
LDA	96.38	96.98	95.78	96.37	99.24	95.78
NMF	96.34	97.26	95.47	96.35	99.19	95.43
LDA2VEC	95.92	96.67	95.21	95.93	99.04	95.19
TOP2VEC	95.80	96.35	95.26	95.80	99.01	95.26
ETM	95.45	95.86	95.05	95.44	98.88	95.05
BERTopic	95.44	95.96	94.91	95.43	98.84	94.91
FLSA	94.41	94.67	94.15	94.38	98.65	94.12

Table 3. TextNetTopics performance results for the WOS dataset, utilizing various topic modeling methods in the T component (150 features). The highest values for each performance metric are highlighted in bold.

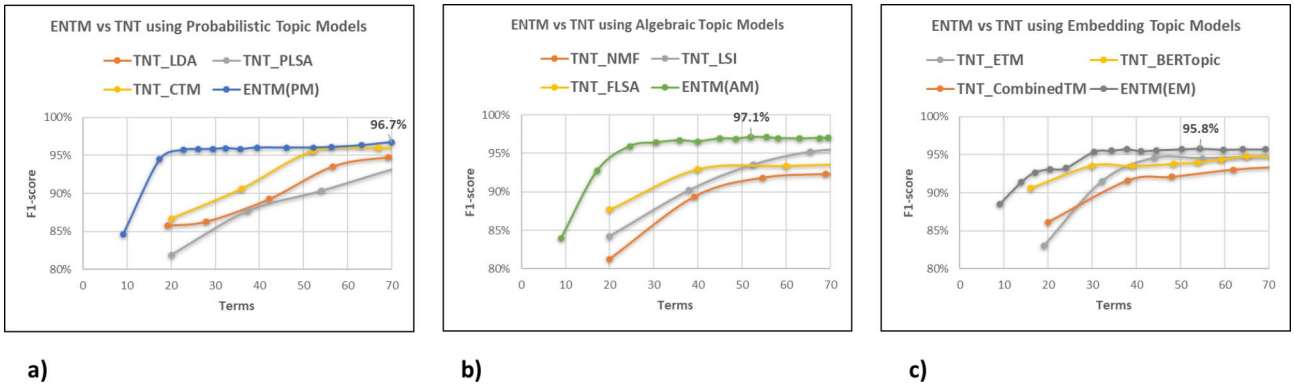


Fig. 9. F1-score performance results for ENTM-TS vs. TextNetTopics on the WOS dataset using: (**a**) Probabilistic topic models, (**b**) Algebraic topic models, (**c**) Embedding topic models.

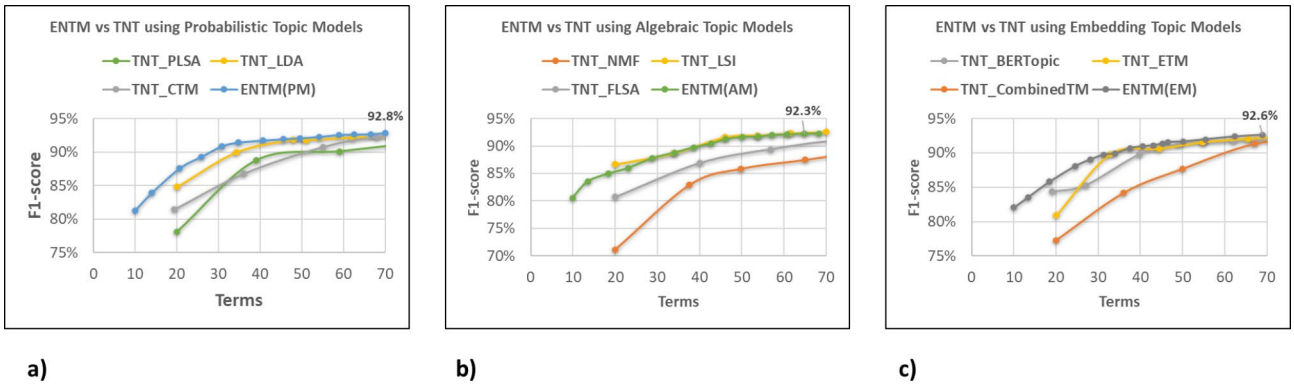


Fig. 10. F1-score performance results for ENTM-TS vs. TextNetTopics on the CAMDA dataset using: (**a**) Probabilistic topic models, (**b**) Algebraic topic models, (**c**) Embedding topic models.

To ensure a fair comparison with the results published by TextNetTopics[3], we selected twenty topics with twenty words per topic, as these settings had demonstrated superior performance in the context of TextNetTopics. The decision to set the group size to ten words per group was based on a detailed rationale provided in the following section. Additionally, we chose a minimum similarity threshold of 0.7 for merging and conducted three iterations of topic merging (f). These parameter choices were empirically determined by the authors, resulting in the best performance observed in their experiments.

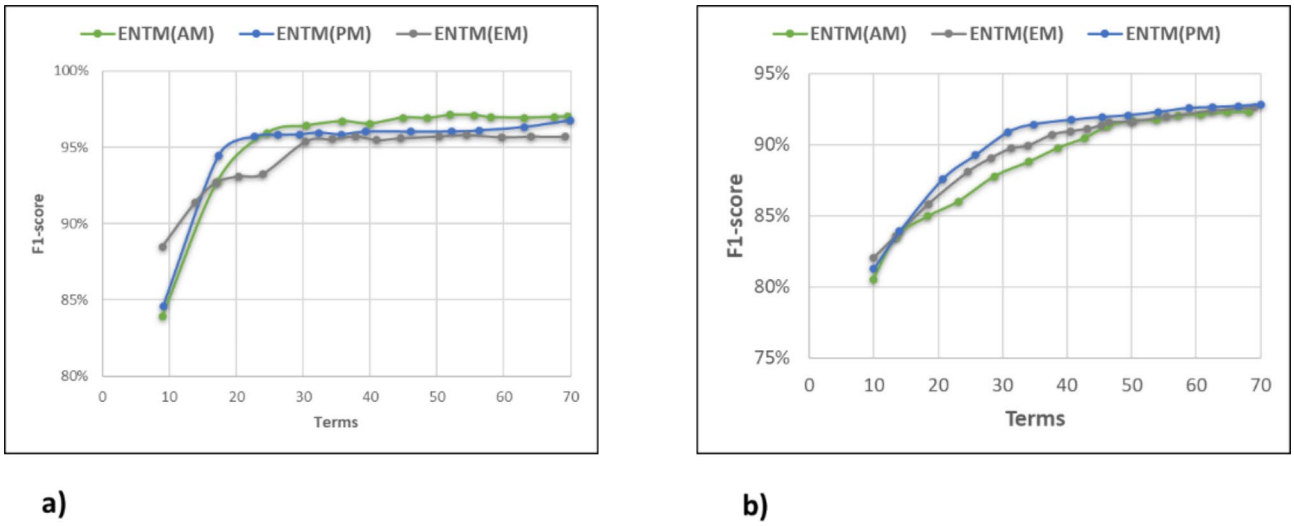


Fig. 11. F1-score performance results for ENTM-TS across the three types of topic models: (**a**) on the WOS Dataset, (**b**) on the CAMDA dataset.

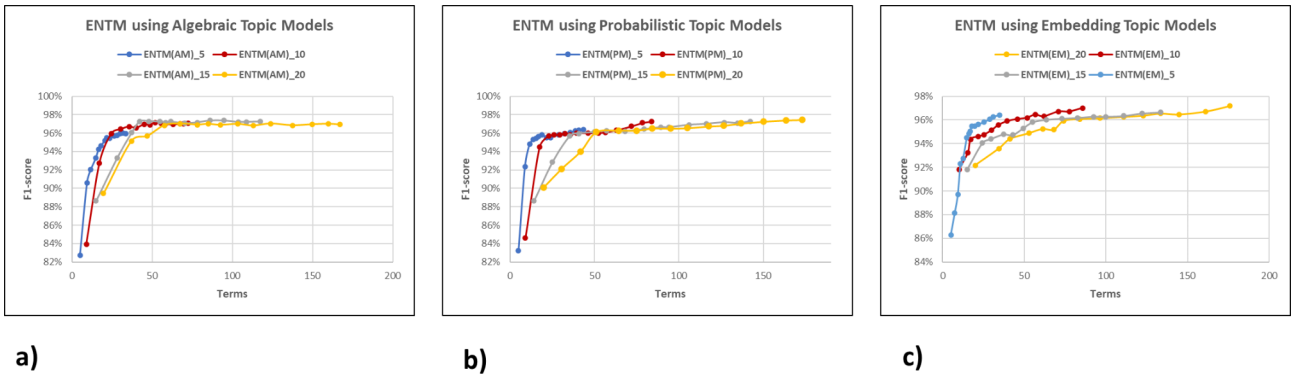


Fig. 12. F1-score performance results for ENTM-TS for various group sizes on the WOS dataset using: (**a**) Algebraic topic models, (**b**) Probabilistic topic models, (**c**) Embedding topic models.

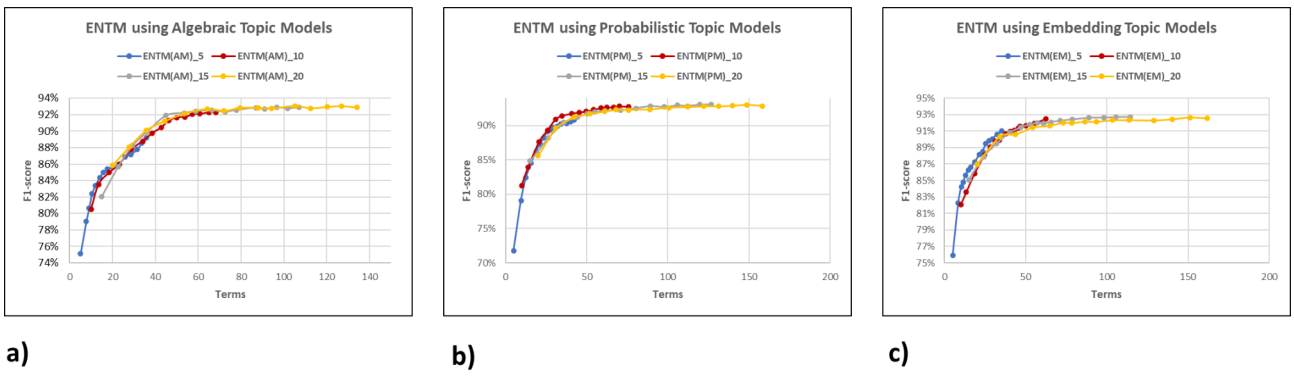


Fig. 13. F1-score performance results for ENTM-TS for various group sizes on the CAMDA dataset using: (**a**) Algebraic topic models, (**b**) Probabilistic topic models, (**c**) Embedding topic models.

Figure 9 illustrates the F1-performance across different term sets in the WOS dataset. Remarkably, we observe a notable improvement when utilizing ENTM, which ensembles multiple topic models, as opposed to TNT, which relies on their equivalent single counterparts.

Extending our analysis to the CAMDA dataset, Fig. 10 illustrates the F1-performance across diverse term sets. Notably, we observe a key turning point, specifically when the number of features reaches 30 for algebraic and embedding-based topic models and 50 for probabilistic models. At this juncture, the performance becomes comparable to that achieved with TNT, which consistently yields the highest performance.

These observations underscore the remarkable ability of ENTM-TS to mitigate the inconsistency in performance exhibited by different topic models. It signifies the effectiveness of our approach in achieving a level of performance that aligns with or surpasses the optimal outcomes obtained with individual topic models, such as those employed by TNT. The ensemble nature of ENTM-TS emerges as a key factor contributing to its efficacy in text classification tasks.

In Fig. 11, a comparative analysis of the three types of Topic Modeling employed within ENTM-TS reveals that the outcomes are dataset-dependent. In the case of the WOS dataset, beyond 25 features, the sequence of performance is observed to be highest for the Algebraic Model, followed by the Probabilistic Model, and then the Embedding Model.

Conversely, for the CAMDA dataset, a dataset-specific trend emerges. Within the range of 20 to 50 features, the Probabilistic Model surpasses the Embedding Model, and the Embedding Model, in turn, outperforms the Algebraic Model. This dataset-dependent variation underscores the nuanced nature of the relationship between feature selection and the specific characteristics of each dataset.

Exploring the impact of group size on ENTM-TS performance

This section investigates the effect of varying the group size on ENTM-TS Performance. In specific, the number of words selected from each group was set to 5, 10, 15, and 20. The remaining parameters were set fixed (similarity threshold = 0.7, topic merging iterations = 3, and the number of topics and words per topic to 20). Results from Fig. 12 for the WOS dataset and Fig. 13 for the CAMDA dataset indicate that accumulating 15 highly ranked groups, each comprising ten words with high XGBoost importance values, yields comparable performance to aggregating the same number of groups with 15 or 20 words. For instance, when we ensemble probabilistic topic modeling algorithms in the WOS dataset, an F1-score of 97% was achieved while reducing the total word subset size (yielded from accumulating groups' words) by 41% and 52% when considering 10 words per group versus 15 and 20 group sizes, respectively. For the CAMDA dataset, the ensemble approach for the probabilistic topic modeling algorithms achieved an F1-score of 93% while reducing the total word subset size by 39% and 55% when setting group size to 10 versus 15 and 20, respectively. These trends were consistent across ensemble algebraic and embedding-based topic models for both datasets.

These findings indicate that by strategically selecting a subset of words from each group, we can achieve comparable results in terms of F1-score while significantly reducing the number of terms considered, thus improving the model's efficiency and interpretability.

Conclusion

This study investigated the impact of using alternative topic modeling methods in the T component, such as LSI, NMF, FLSA, LDA, CTM, PLSA, LDA2VEC, TOP2VEC, BERTopic, CombinedTM, and ETM on TextNetTopics performance. From the results obtained, incorporating TextNetTopics with the LSI enables us to represent documents with a reduced number of highly representative features that provide a similar ability to others in discriminating the two classes in binary classification. For instance, in the DILI classification problem, to provide the same F1-score of 93.00%, LSI uses approximately 66%, 60%, 55%, and 45% fewer features (terms) than PLSA, CombinedTM, NMF, and ETM, respectively, while providing similar performance measures (e.g., accuracy, AUC).

In addition, we have explored the significant list of extracted topics by each topic modeling method. From the presented results, we have noticed that when we increase the feature set (accumulated top-ranked topics) for each method, a notable improvement in all performance measures yields; this indicates the importance of TextNetTopics, which can discover significant topics that highly discriminate the classes in this classification problem.

However, TextNetTopics incorporated with an individual topic model exhibits inconsistency in performance across various datasets. Therefore, this study introduced ENTM-TS (Ensemble Topic Modeling for Topic Selection), a novel approach designed to enhance text classification over TextNetTopics by creating topics with enhanced discriminative power intended as input features for training machine learning algorithms. By integrating multiple topic models using the Grouping, Scoring, and Modeling approach, ENTM-TS demonstrates a remarkable ability to mitigate the variability introduced by individual topic modeling methods. Our comprehensive evaluation, conducted on the WOS and CAMDA datasets, showcases the effectiveness of ENTM-TS in surpassing or aligning with optimal outcomes obtained from individual topic models.

A noteworthy constraint of ENTM-TS is its requirement for users to specify certain parameters manually. Specifically, users must provide predetermined values for several parameters, including the number of topics to be extracted, the desired number of words per topic, the number of iterations for the topic merging process, and the number of words to be utilized from each group for normalizing group sizes. While this empowers users with control over the modeling process, it introduces a potential hurdle for those seeking a more automated or adaptive approach. In future work, we will explore implementing automated methods or heuristics for optimal parameter tuning within ENTM-TS to address the mentioned limitation and enhance the user experience.

Data availability

The datasets analyzed during the current study are available in the GitHub repository, https://github.com/malikyousef/ENTM-TS.

Code availability

The KNIME implementation workflows for NLP preprocessing tasks, TextNetTopics, and ENTM-TS tools, along with the Google Colab notebook implementing the topic models used in this study, are accessible on the KNIME Hub at https://hub.knime.com/malik/spaces/ENTM-TS/~siq1snGdh7BUI5QB/ or on GitHub https://github.com/malikyousef/ENTM-TS.

Received: 4 January 2024; Accepted: 23 September 2024
Published online: 09 October 2024

References

1. Kadhim, A. I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **52**(1), 273–292. https://doi.org/10.1007/s10462-018-09677-1 (2019).
2. Onan, A., Korukoglu, S. & Bulut, H. LDA-based topic modelling in text sentiment classification: An empirical analysis. *Int. J. Comput. Linguist. Appl.* **7**(1), 101–119 (2016).
3. Yousef, M. & Voskergian, D. TextNetTopics: Text classification based word grouping as topics and topics' scoring. *Front. Genet.* **13**, 893378. https://doi.org/10.3389/fgene.2022.893378 (2022).
4. Blair, S. J., Bi, Y. & Mulvenna, M. D. Aggregated topic models for increasing social media topic coherence. *Appl. Intell.* **50**(1), 138–156. https://doi.org/10.1007/s10489-019-01438-z (2020).
5. Belford, M. & Greene, D. Ensemble topic modeling using weighted term co-associations. *Expert Syst. Appl.* **161**, 113709. https://doi.org/10.1016/j.eswa.2020.113709 (2020).
6. Belford, M., MacNamee, B. & Greene, D. Stability of topic modeling via matrix factorization. *Expert Syst. Appl.* **91**, 159–169. https://doi.org/10.1016/j.eswa.2017.08.047 (2018).
7. Blair, S. J., Bi, Y. & Mulvenna, M. D. Increasing topic coherence by aggregating topic models. In *Knowledge Science, Engineering and Management. Lecture Notes in Computer Science* Vol. 9983 (eds Lehner, F. & Fteimi, N.) 69–81 (Springer International Publishing, Cham, 2016). https://doi.org/10.1007/978-3-319-47650-6_6.
8. Luo, L. & Li, L. Defining and evaluating classification algorithm for high-dimensional data based on latent topics. *PLoS ONE* **9**(1), e82119. https://doi.org/10.1371/journal.pone.0082119 (2014).
9. Al-Salemi, B., Ab Aziz, M. J. & Noah, S. A. LDA-AdaBoost. MH: Accelerated AdaBoost. MH based on latent Dirichlet allocation for text categorization. *J. Inf. Sci.* **41**(1), 27–40. https://doi.org/10.1177/0165551514551496 (2015).
10. Alhaj, F., Al-Haj, A., Sharieh, A. & Jabri, R. Improving Arabic cognitive distortion classification in twitter using BERTopic. *IJACSA* **13**(1). https://doi.org/10.14569/IJACSA.2022.0130199 (2022).
11. Glazkova, A. Using topic modeling to improve the quality of age-based text classification. In *CEUR Workshop Proceedings* 92–97 (2021).
12. Rijcken, E. et al. Topic modeling for interpretable text classification from EHRs. *Front. Big Data* **5**, 846930. https://doi.org/10.3389/fdata.2022.846930 (2022).
13. Zrigui, M., Ayadi, R., Mars, M. & Maraoui, M. Arabic text classification framework based on latent Dirichlet allocation. *CIT* **20**(2). https://doi.org/10.2498/cit.1001770 (2012).
14. Zhang, Z., Phan, X.-H. & Horiguchi, S. An efficient feature selection using hidden topic in text categorization. In *22nd International Conference on Advanced Information Networking and Applications—Workshops (aina workshops 2008)* 1223–1228 (IEEE, Ginowan, 2008). https://doi.org/10.1109/WAINA.2008.137 (2008).
15. Tasci, S. & Gungor T. LDA-based keyword selection in text categorization. In *2009 24th International Symposium on Computer and Information Sciences*, 230–235 (IEEE, Guzelyurt, 2009). https://doi.org/10.1109/ISCIS.2009.5291818.
16. Al-Salemi, B., Ayob, M., Noah, S. A. M. & Ab Aziz, M. J. 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)* 1–6 (IEEE, Langkawi, 2017). https://doi.org/10.1109/ICEEI.2017.8312411.
17. Mo, Y., Kontonatsios, G. & Ananiadou, S. Supporting systematic reviews using LDA-based document representations. *Syst. Rev.* **4**(1), 172. https://doi.org/10.1186/s13643-015-0117-0 (2015).
18. Aguiar, A., Silveira, R., Furtado, V., Pinheiro, V. & Neto, J. A. M. Using topic modeling in classification of Brazilian lawsuits. In *Computational Processing of the Portuguese Language. Lecture Notes in Computer Science* Vol. 13208 (eds Pinheiro, V. et al.) 233–242 (Springer International Publishing, Cham, 2022). https://doi.org/10.1007/978-3-030-98305-5_22.
19. Yousef, M., Kumar, A. & Bakir-Gungor, B. Application of biological domain knowledge based feature selection on gene expression data. *Entropy* **23**(1), 2. https://doi.org/10.3390/e23010002 (2020).
20. Yousef, M., Allmer, J., İnal, Y. & Gungor, B. B. G-S-M: A comprehensive framework for integrative feature selection in omics data analysis and beyond. https://doi.org/10.1101/2024.03.30.585514 (2024).
21. Kuzudisli, C., Bakir-Gungor, B., Bulut, N., Qaqish, B. & Yousef, M. Review of feature selection approaches based on grouping of features. *PeerJ* **11**, e15666. https://doi.org/10.7717/peerj.15666 (2023).
22. Voskergian, D., Bakir-Gungor, B. & Yousef, M. TextNetTopics Pro, a topic model-based text classification for short text by integration of semantic and document-topic distribution information. *Front. Genet.* **14**, 1243874. https://doi.org/10.3389/fgene.2023.1243874 (2023).
23. Yousef, M., Jung, S., Showe, L. C. & Showe, M. K. Recursive Cluster Elimination (RCE) for classification and feature selection from gene expression data. *BMC Bioinform.* **8**(1), 144. https://doi.org/10.1186/1471-2105-8-144 (2007).
24. Yousef, M. et al. Recursive cluster elimination based rank function (SVM-RCE-R) implemented in KNIME. *F1000Res* **9**, 1255. https://doi.org/10.12688/f1000research.26880.2 (2021).
25. Yousef, M., Jabeer, A. & Bakir-Gungor, B. SVM-RCE-R-OPT: Optimization of scoring function for SVM-RCE-R. In *Database and Expert Systems Applications—DEXA 2021 Workshops Communications in Computer and Information Science* Vol. 1479 (eds Kotsis, G. et al.) 215–224 (Springer International Publishing, Cham, 2021). https://doi.org/10.1007/978-3-030-87101-7_21.
26. Yousef, M., Ketany, M., Manevitz, L., Showe, L. C. & Showe, M. K. Classification and biomarker identification using gene network modules and support vector machines. *BMC Bioinform.* **10**(1), 337. https://doi.org/10.1186/1471-2105-10-337 (2009).
27. Yousef, M., Abdallah, L. & Allmer, J. maTE: Discovering expressed interactions between microRNAs and their targets. *Bioinform.* **35**(20), 4020–4028. https://doi.org/10.1093/bioinformatics/btz204 (2019).
28. Yousef, M., Ülgen, E. & Uğur Sezerman, O. CogNet: Classification of gene expression data based on ranked active-subnetwork-oriented KEGG pathway enrichment analysis. *PeerJ Comput. Sci.* **7**, e336. https://doi.org/10.7717/peerj-cs.336 (2021).
29. Yousef, M. et al. miRcorrNet: Machine learning-based integration of miRNA and mRNA expression profiles, combined with feature grouping and ranking. *PeerJ* **9**, e11458. https://doi.org/10.7717/peerj.11458 (2021).

30. Yousef, M., Goy, G. & Bakir-Gungor, B. miRModuleNet: Detecting miRNA-mRNA regulatory modules. *Front. Genet.* **13**, 767455. https://doi.org/10.3389/fgene.2022.767455 (2022).
31. Yousef, M., Sayıcı, A. & Bakir-Gungor, B. Integrating gene ontology based grouping and ranking into the machine learning algorithm for gene expression data analysis. In *Database and Expert Systems Applications—DEXA 2021 Workshops. Communications in Computer and Information Science* Vol. 1479 (eds Kotsis, G. et al.) 205–214 (Springer International Publishing, Cham, 2021). https://doi.org/10.1007/978-3-030-87101-7_20.
32. Yousef, M., Ozdemir, F., Jaaber, A., Allmer, J. & Bakir-Gungor, B. PriPath: Identifying dysregulated pathways from differential gene expression via grouping, scoring and modeling with an embedded machine learning approach. In Review, preprint, Apr. 2022. https://doi.org/10.21203/rs.3.rs-1449467/v1.
33. Qumsiyeh, E., Showe, L. & Yousef, M. GediNET for discovering gene associations across diseases using knowledge based machine learning approach. *Sci. Rep.* **12**(1), 19955. https://doi.org/10.1038/s41598-022-24421-0 (2022).
34. Jabeer, A., Temiz, M., Bakir-Gungor, B. & Yousef, M. miRdisNET: Discovering microRNA biomarkers that are associated with diseases utilizing biological knowledge-based machine learning. *Front. Genet.* **13**, 1076554. https://doi.org/10.3389/fgene.2022.1076554 (2023).
35. Ersoz, N. S., Bakir-Gungor, B. & Yousef, M. GeNetOntology: Identifying affected gene ontology groups via grouping, scoring and modelling from gene expression data utilizing biological knowledge based machine learning. *Front. Genet.* **14**, 1139082 (2023).
36. Unlu Yazici, M., Marron, J. S., Bakir-Gungor, B., Zou, F. & Yousef, M. Invention of 3Mint for feature grouping and scoring in multi-omics. *Front. Genet.* **14**, 1093326. https://doi.org/10.3389/fgene.2023.1093326 (2023).
37. Qumsiyeh, E., Salah, Z. & Yousef, M. miRGediNET: A comprehensive examination of common genes in miRNA-Target interactions and disease associations: Insights from a grouping-scoring-modeling approach. *Heliyon* **9**(12), e22666. https://doi.org/10.1016/j.heliyon.2023.e22666 (2023).
38. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**(6755), 788–791. https://doi.org/10.1038/44565 (1999).
39. Landauer, T. K., Foltz, P. W. & Laham, D. An introduction to latent semantic analysis. *Discourse Process.* **25**(2–3), 259–284. https://doi.org/10.1080/01638539809545028 (1998).
40. Rijcken, E., Scheepers, F., Mosteiro, P., Zervanou, K., Spruit, M. & Kaymak, U. A comparative study of fuzzy topic models and LDA in terms of interpretability. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–8 (IEEE, Orlando, 2021). https://doi.org/10.1109/SSCI50451.2021.9660139.
41. Hofmann, T. Probabilistic latent semantic analysis. https://doi.org/10.48550/ARXIV.1301.6705 (2013).
42. Blei, D. M. & Lafferty, J. D. A correlated topic model of Science. *Ann. Appl. Stat.* **1**(1). https://doi.org/10.1214/07-AOAS114 (2007).
43. Moody, C. E. Mixing Dirichlet topic models and word embeddings to make lda2vec. https://doi.org/10.48550/ARXIV.1605.02019 (2016).
44. Angelov, D. Top2Vec: Distributed representations of topics. https://doi.org/10.48550/ARXIV.2008.09470 (2020).
45. Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. https://doi.org/10.48550/ARXIV.2203.05794 (2022).
46. Bianchi, F., Terragni, S. & Hovy, D. Pre-training is a hot topic: Contextualized Document embeddings improve topic coherence. https://doi.org/10.48550/ARXIV.2004.03974 (2020).
47. Dieng, A. B., Ruiz, F. J. R. & Blei, D. M. Topic modeling in embedding spaces. https://doi.org/10.48550/ARXIV.1907.04907 (2019).
48. Alghamdi, R. & Alfalgi, K. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, **6**(1) (2015).
49. Kherwa, P. & Bansal, P. Topic modeling: A comprehensive review. *ICST Trans. Scalable Inf. Syst.* 159623. https://doi.org/10.4108/eai.13-7-2018.159623 (Jul.2018).
50. Landauer, T. K., Laham, D., Rehder, B. & Schreiner, M. E. How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In *Proceedings of the 19th Annual Meeting of the Cognitive Science Society*, 412–417 (1997).
51. Dumais, S. T. Latent semantic analysis. *Annu. Rev. Inf. Sci. Technol.* **38**(1), 188–230 (2004).
52. Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 50–57 (1999).
53. Rijcken, E., Zervanou, K., Mosteiro, P., Spruit, M., Scheepers, F. & Kaymak, U. A performance evaluation of topic models based on fuzzy latent semantic analysis (2022).
54. Blei, D. M. Probabilistic topic models. *Commun. ACM* **55**(4), 77–84. https://doi.org/10.1145/2133806.2133826 (2012).
55. Mohammed, S. H. & Al-augby, S. Lsa & lda topic modeling classification: Comparison study on e-books. *Indones. J. Electr. Eng. Comput. Sci.* **19**(1), 353–362 (2020).
56. Blei, D. M. *Probabilistic Models of Text and Images* (University of California, Berkeley, 2004).
57. Mifrah, S. & Benlahmar, E. H. Topic modeling coherence: A comparative study between LDA and NMF models using COVID'19 corpus. *Int. J. Adv. Trends Comput. Sci. Eng.* 5756–5761 (2020).
58. GitHub—ddangelov/Top2Vec: Top2Vec learns jointly embedded topic, document and word vectors. Accessed: Nov. 14, 2022. [Online]. Available: https://github.com/ddangelov/Top2Vec
59. Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S. & Barnes, L. E. HDLTex: Hierarchical Deep Learning for Text Classification. https://doi.org/10.48550/ARXIV.1709.08267 (2017).
60. "malik/TextNetTopics_TM," KNIME Community Hub. Accessed: Feb. 21, 2023. [Online]. Available: https://hub.knime.com/malik/spaces/TextNetTopics_TM/latest/
61. Yousef, M. TextNetTopics UTILIZING VARIOUS TOPIC MODELING METHODS. Feb. 21, 2023. Accessed: Feb. 21, 2023. [Online]. Available: https://github.com/malikyousef/TextNetTopics_TM
62. "GitHub—yedivanseven/PLSA: Probabilistic Latent Semantic Analysis." Accessed: Nov. 14, 2022. [Online]. Available: https://github.com/yedivanseven/PLSA
63. Lee, M. "tomotopy". Dec. 17, 2022. Accessed: Dec. 18, 2022. [Online]. Available: https://github.com/bab2min/tomotopy
64. Newman, D., Asuncion, A., Smyth, P. & Welling, M. Distributed algorithms for topic models. *J. Mach. Learn. Res.* **10**(8) (2009).
65. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V. & Thirion, B. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
66. Rijcken, E. Fuzzy topic modeling—Methods derived from Fuzzy Latent Semantic Analysis. Dec. 16, 2022. Accessed: Dec. 18, 2022. [Online]. Available: https://github.com/ERijck/FuzzyTM.
67. Rijcken, E., Mosteiro, P., Zervanou, K., Spruit, M., Scheepers, F. & Kaymak, U. FuzzyTM: A software package for fuzzy topic modeling. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–8 (IEEE, Padua, 2022). https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882661.
68. Raw, N. Lda2vec-Tensorflow. Jun. 27, 2022. Accessed: Jul. 17, 2022. [Online]. Available: https://github.com/nateraw/Lda2vec-Tensorflow.
69. "GitHub—MaartenGr/BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics." Accessed: Nov. 10, 2022. [Online]. Available: https://github.com/MaartenGr/BERTopic.
70. "Contextualized Topic Models." MilaNLP, Dec. 24, 2022. Accessed: Dec. 26, 2022. [Online]. Available: https://github.com/MilaNLProc/contextualized-topic-models.
71. Dieng, A. B. ETM. Dec. 17, 2022. Accessed: Dec. 26, 2022. [Online]. Available: https://github.com/adjidieng/ETM.

72. Karami, A., Gangopadhyay, A., Zhou, B. & Kharrazi, H. Fuzzy approach topic discovery in health and medical corpora. https://doi.org/10.48550/ARXIV.1705.00995 (2017).

Author contributions

D.V. and M.Y. played integral roles in conceptualization, formal analysis, investigation, methodology, writing the original draft, validation, visualization, and software development. R.J. took part in project administration, validation, writing (including review and editing), and supervision.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.V. or M.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.