Journal of Theoretical and Applied Information Technology
15th June 2025. Vol.103. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN INVESTIGATION OF DYNAMIC TOPIC MODELING FOR REAL-TIME AND EVOLVING TEXTUAL DATA USING DTM, BERTOPIC, RECURRENT NEURAL NETWORKS AND PROPOSED HYBRID DTM WITH RNN ALGORITHMS

[1]C.B.PAVITHRA, [2]DR.J.SAVITHA

[1]Research Scholar, Department of Information Technology,
Dr.N.G.P. Arts & Science College, Coimbatore, Tamilnadu, India.
[2]Professor, Department of Information Technology,
Dr.N.G.P. Arts & Science College, Coimbatore, Tamilnadu, India.
E-mail: c.b.pavithramsc2004@gmail.com[1], savithaj@drngpasc.ac.in[2]

ABSTRACT

Given the dynamic nature of textual data, Dynamic Topic Modeling has become an effective real-time analysis tool for streams of textual data. The objective of this study is to present a thorough review of different dynamic topic modeling strategies, such as advanced neural network-based methods like Recurrent Neural Networks (RNN), recent methodology like BERTopic, and traditional approaches like DTM. It also looks at the possible advantages and difficulties of combining RNN and DTM in a hybrid framework. We explore the effectiveness of these techniques in capturing temporal dynamics, identifying changing subjects, and offering insights into the underlying structures of the data through empirical evaluations on real-world textual datasets. Using the "Advanced Topic Modeling for Research Articles 2.0" dataset, this study assesses the methods according to a number of criteria, including accuracy, recall, precision, coherence, perplexity, and F-score. This research also assesses the subject modeling performance, scalability, and flexibility of our hybrid DTM and RNN strategy in relation to real-time and dynamic textual data, in comparison with other methods. The outcomes of our trials highlight the benefits of this hybrid strategy and offer insightful information to practitioners and researchers who want to use dynamic topic modeling for textual data analysis that is dynamic and real-time.

Keywords: *Dynamic Topic Modeling, Real-Time Data Analysis, Textual Data Streams, DTM, BERTopic, Recurrent Neural Networks, RNN, Hybrid Models, Natural Language Processing and Text Mining.*

1. INTRODUCTION

Effective analytic approaches are now critical in an era marked by an extraordinary inflow of textual data from a variety of sources, including social media, news articles, and online forums. Due to the dynamic and evolving nature of these data streams, traditional static topic modeling techniques like Latent Dirichlet Allocation (LDA) are insufficient [1]. To tackle this problem, dynamic topic modeling (DTM) has emerged as a promising approach that enables real-time meaningful topic extraction from continuously evolving textual data. The temporal dynamics present in textual data streams can be captured by DTM algorithms, making it possible to identify subjects that change over time. Nonetheless, with the introduction of fresh approaches and the incorporation of cutting-edge technology like deep learning, the field of dynamic topic modeling has seen tremendous developments in the last few years [2][3]. The purpose of this study is to present a thorough review of several dynamic topic modeling strategies, from conventional DTM approaches to cutting-edge approaches like BERTopic and Recurrent Neural Networks (RNN). This study examines the possible advantages and difficulties of integrating DTM and RNN in a hybrid framework that combines the best features of both approaches. Hybrid approaches in dynamic topic modeling, particularly the integration of Dynamic Topic Modeling (DTM) with Recurrent Neural Networks (RNN), are motivated by several key factors:

- ***Complementary Strengths:*** DTM is useful for extracting changing topics and capturing the temporal dynamics of textual data streams. RNNs, on the other hand, are excellent at capturing long-term dependencies and modeling sequential data. Our goal in merging these two approaches is to take advantage of their complimentary

qualities in order to improve the precision and effectiveness of topic modeling for dynamic, real-time text data.

- ***Improved Flexibility:*** More modeling freedom is available for complex data structures when using hybrid methodologies. RNNs can capture fine-grained temporal connections inside individual texts, but DTM is better suited to capture general subject change across time. The integration of these approaches facilitates a more sophisticated comprehension of the dynamics of textual material.
- ***Enhanced Adaptability:*** Textual data from the real world frequently demonstrates a variety of traits, such as differing lengths, subtle linguistic expressions, and changing subjects. By fusing the adaptability of RNNs with the flexibility of DTM, hybrid techniques offer a more flexible framework for managing such complexity, enhancing the resilience of topic modeling algorithms.
- ***Better Performance:*** We hope to get around some of the drawbacks of conventional dynamic topic modeling methods, namely DTM's dependence on predetermined discrete time intervals, by combining DTM with RNNs. RNNs' dynamic properties enable ongoing learning and adaptation to shifting data distributions, which may enhance their ability to capture developing subjects.
- ***Scalability and Efficiency:*** Scalability and efficiency benefits are possible using hybrid techniques, especially when dealing with large-scale textual data streams. Hybrid models may handle enormous volumes of data more effectively by utilizing the distributed computing methods included into DTM frameworks and the parallel processing powers of RNNs. This allows for real-time topic modeling at scale.

The need for this paper arises from the growing volume and velocity of textual data in real-time platforms such as news feeds and social media, where topic dynamics evolve rapidly. Traditional models like LDA or static topic models fail to adapt to these temporal and contextual changes. While DTM captures temporal transitions and RNN captures sequential dependencies, there exists a research gap in a hybrid approach that combines both strengths for robust, scalable, and adaptive topic modeling. The addressed problem is: How can we improve the quality, coherence, and temporal relevance of topic models in real-time data streams?

The rest of this essay is structured as follows: A summary of conventional Dynamic Topic Modeling methods is given in Section 2, with an emphasis on the related works. In Section 3, discussed Proposed Methodology are Hybrid Dynamic Topic Modeling (DTM) And Recurrent Neural Networks (RNN) For Topic Modeling. A comparative study of the methods outlined in Section 5 is provided, along with a scalability and performance metrics evaluation. The work is finally brought to a close in Section 6, which summarizes the main conclusions and explores possible directions for further study in the area of dynamic topic modeling for real-time and developing textual data analysis.

2. RELATED WORKS

Topic modeling is a statistical method widely used in text mining and natural language processing to uncover recurring themes or subjects within a collection of documents. Its primary aim is to automatically identify patterns in how words co-occur across texts, revealing underlying semantic structures. Typically, this involves creating a document-term matrix where terms (words or phrases) form columns and documents form rows. Latent Dirichlet Allocation (LDA) is a prominent algorithm in topic modeling, assuming documents consist of mixtures of topics, each represented as a probability distribution over terms. LDA effectively identifies consistent themes by estimating word distributions and topic proportions for each document. Evaluating the semantic coherence and similarity of top words within each topic are common practices in assessing topic model quality. Beyond traditional static methods, newer approaches such as BERTopic and dynamic techniques like Dynamic Topic Modeling (DTM) have emerged to handle large-scale textual data and capture temporal shifts in topics over time [4][5]. Topic modeling finds applications in diverse fields including document clustering, information retrieval, trend analysis, and recommendation systems, making it a valuable tool for analyzing and understanding textual content.

Dynamic Topic Modeling (DTM) has emerged as a powerful technique for analyzing text data that evolves over time. DTM extends traditional topic modeling methods like Latent Dirichlet

Allocation (LDA) to capture temporal dynamics in document collections. Introduced by ***Blei and Lafferty (2006)*** [6], DTM models how topics evolve over time by incorporating time-dependent parameters into the generative process of topic modeling. DTM has been applied in various domains including analyzing news archives ***(A. Ahmed et al., 2010)*** [7] and tracking topics in short text and original document ***(Li X et al., 2017)*** [8], demonstrating its utility in capturing temporal changes in topics. Bertopic represents a recent advancement in topic modeling that leverages contextual embeddings from pre-trained language models like BERT. Proposed by ***Gens et al. (2020)*** [9], Bertopic uses BERT embeddings to represent documents and performs clustering based on semantic similarity, thereby improving the quality of topic representations. Bertopic has been applied to tasks such as document clustering ***(Gens et al., 2020)*** and semantic search ***(Reimers and Gurevych, 2019)*** [10], showcasing its effectiveness in capturing nuanced semantic relationships among documents. Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data, making them suitable for tasks involving natural language processing (NLP). Introduced by ***Hochreiter and Schmidhuber (1997)*** [11], RNNs maintain a state that evolves as they process sequences, enabling them to capture dependencies over time. Several notable advancements in recurrent neural networks (RNNs) have been proposed in recent literature. ***Bacciu et al. (2020)*** [12] explored methods to enhance the resilience of RNNs through dropout techniques, aiming to improve their robustness in handling complex data sequences. ***Dieng et al. (2017)*** [13] introduced TopicRnn, a novel RNN architecture designed to capture long-range semantic dependencies, thereby enhancing its effectiveness in tasks requiring nuanced understanding of textual contexts. ***Qin et al. (2017)*** [14] proposed a dual-stage attention-based RNN specifically tailored for time series prediction, leveraging attention mechanisms to focus on relevant temporal features. ***Mikolov et al. (2010)*** [15] contributed significantly to the field with their work on RNN-based language models, which have since been pivotal in natural language processing tasks such as speech recognition and machine translation. Additionally, ***Kudinov et al. (2016)*** [16] combined RNNs with probabilistic topic modeling to develop a hybrid language model, aiming to integrate semantic understanding with sequential data processing, showcasing a promising direction for improving text generation and understanding algorithms. These studies collectively highlight the diverse applications and ongoing innovations in RNNs, shaping their evolution and expanding their utility across various domains of artificial intelligence and machine learning. Recent research has explored hybrid approaches that combine DTM with deep learning techniques such as RNNs to enhance the modeling of temporal dynamics and sequential dependencies in evolving text data.

Existing topic modeling methods are limited in their ability to capture both temporal topic evolution and sequential dependencies within dynamic textual data. This creates a need for hybrid approaches that can adapt to real-time data streams with enhanced interpretability and accuracy.

Previous studies have explored either traditional DTM for temporal modeling (e.g., Blei & Lafferty, 2006) or used neural methods like BERTopic and RNNs for semantic representation. While each shows strength in specific aspects, limitations remain—DTM struggles with fine-grained sequence modeling and RNNs lack explicit temporal topic tracking. Our study differs by proposing a hybrid DTM-RNN architecture that captures both temporal evolution and sequential word-level patterns. Our findings demonstrate superior performance in coherence, perplexity, and classification accuracy compared to standalone models, offering a holistic improvement in topic modeling for evolving text streams.

3. PROPOSED METHODOLOGY (HYBRID DYNAMIC TOPIC MODELING (DTM) AND RECURRENT NEURAL NETWORKS (RNN) FOR TOPIC MODELING)

In natural language processing, topic modeling is an essential tool that helps reveal hidden topics in a corpus of textual data. The suggested Hybrid Dynamic Topic Model (DTM) and Recurrent Neural Network (RNN) methodology combines the advantages of two different approaches into a more complete topic modeling solution. The Dynamic Topic Model (DTM) provides a useful way to model how topics change over time by capturing the temporal evolution of topics within

a corpus. DTM offers insights into the dynamic nature of topics within the text data by modeling the transition probabilities between topics across time slices and estimates topic proportions for each document. Meanwhile, Recurrent Neural Networks (RNNs) are excellent at recognizing sequential dependencies in data streams. Deeper comprehension of the semantic structure of documents is made possible by RNNs' ability to train representations that capture the contextual relationships between words in the context of text data. The complementing nature of DTM and RNN techniques is utilized when integrating them into a hybrid architecture. Through the integration of DTM's temporal dynamics and RNNs' sequential dependencies, the hybrid technique provides a more comprehensive comprehension of the latent subjects included in text data. Through improved topic modeling made possible by this integration, applications in text mining, social media analysis, document classification, and content analysis as well as information retrieval are made possible.

Recurrent Neural Networks (RNN) and Dynamic Topic Modeling (DTM) provide unique benefits for encoding temporal dynamics and sequential dependencies, respectively, in textual data. By combining these approaches into a hybrid framework, topic modeling is approached comprehensively while utilizing the advantages of each method. Here, we outline an algorithm for topic modeling using Hybrid DTM-RNN, along with step-by-step formulations and technical details.

Algorithm for Proposed Hybrid DTM and RNN for Topic Modeling:

Step 1: Preprocessing

Input: Given a corpus of text documents $D=\{d_1,d_2,...,d_N\}$:

Tokenization: Tokenization splits each document into individual tokens (words, subwords, or characters). For each document d_i, tokenize it into a sequence of tokens:

$$Tokenize(d_i)=[w_{i1},w_{i2},...,w_{iLi}]$$

Where $L_i$is the number of tokens in document d_i.

Normalization: Normalize the tokens by converting them to lowercase to ensure consistency:

$$Normalize\ (w_{ij})=lowercase(w_{ij})$$

Stopword Removal: Remove common stopwords that do not carry much semantic meaning:

$$\text{Remove Stop words}\ (w_{ij}) + \begin{cases} w_{ij} & \text{if } w_{ij} \text{ is not a stopword} \\ \text{null} & \text{, otherwise} \end{cases}$$

Punctuation Removal: Remove punctuation marks from the tokens:

$$\text{Remove Punctuation words}\ (w_{ij}) + \begin{cases} w_{ij} & \text{if } w_{ij} \text{ is not a punctuation mark} \\ \text{null} & \text{, otherwise} \end{cases}$$

Stemming or Lemmatization: Reduce inflected words to their base or root form to normalize variations: Stem(w_{ij}) or Lemmatize(w_{ij})

Handling Numerical Data: Convert numerical tokens into a standard representation:

$$\text{Numerical Handling}\ (w_{ij}) + \begin{cases} \text{number} & \text{if } w_{ij} \text{ is a number} \\ w_{ij} & \text{, otherwise} \end{cases}$$

Join Tokens: Reconstruct the preprocessed tokens into a single string:

$$Joined\ Tokens\ (d_i)= \text{Join}(\text{Preprocessed}(d_i))$$

Step 2: Train the Recurrent Neural Network (RNN)

Input: Given a corpus of documents $D=\{d1,d2,...,dN\}$ and the number of time slices T:

1. Define the term-document matrix: For each document d_i, create a count vector n_i representing the frequency of each term in d_i. Construct a term-document matrix W, where each row represents a document and each column represents a term. The element W_{ij} represents the count of term j in document i.

2. ***Learn topic proportions over time:*** For each time slice t, estimate the topic proportions $P(z_t \mid d_i)$ for each document d_i:

$$P(z_t \mid d_i)= P(z_t)\cdot P(d_i|z_t) \ / \ \sum^{K}_{K=1}P(z_t) \mid P(d_i.z_t)$$

Where, $P(z_t)$ is the prior distribution of topics at time $P(d_i \mid z_t)$ is the likelihood of document d_i given topic z_t.

3. Estimate topic evolution over time: Model the transition probabilities between topics at adjacent time slices: $P(z_{t+1}| z_t)$. This can be estimated using various methods such as Markov chain

Monte Carlo (MCMC) sampling, variation inference, or Gibbs sampling.

4. Optimization: Use an optimization algorithm (e.g., gradient descent) to maximize the likelihood of the observed data given the model parameters.

The objective function to maximize is the log-likelihood of the observed data:

$$L(\Theta)=\sum^{N}{}_{i=1}\sum^{T}{}_{t=1} logP(d_i|z_t)P(z_t)$$

5. Model parameters: Θ represents the parameters of the DTM, including topic distributions, topic proportions, and transition probabilities.

6. Training: Train the DTM using an iterative optimization process until convergence is reached. This involves updating the parameters Θ iteratively to maximize the likelihood of the observed data.

Estimating topic proportions and topic evolution can be done using a variety of inference techniques, including Gibbs sampling and variational inference. Regularization approaches, like transition probabilities or priors on topic distributions, can be used to avoid overfitting. The ideal number of subjects and time slices can be ascertained by applying model selection approaches. This algorithm describes the procedures for training the Dynamic Topic Model (DTM), which includes modeling the evolution of topics and predicting topic proportions across time. Depending on the optimization algorithm and inference method selected, different implementation details may apply.

Step 3: Train the Recurrent Neural Network (RNN)

Training the Recurrent Neural Network (RNN) involves processing sequential data to learn representations capturing temporal dependencies. Here's the formula for training the RNN:

Input: Given a corpus of text documents $D=\{d_1,d_2,...,d_N\}$ represented as sequences of tokens:

1.Tokenization: Tokenize each document d_i into a sequence of tokens $X_i=\{x_{i1},x_{i2},...,x_{iLi}\}$, where L_i is the length of document d_i.

2. Embedding: Map each token x_{ij} to its distributed representation using an embedding matrix

$$\text{Embed}(x_{ij})=eij=E.1x_{ij}$$

Where $1x_{ij}$ is a one-hot vector representation of token x_{ij} and e_{ij} is its embedding.

3. Forward pass: Propagate the embedded tokens through the recurrent layers:

$$h_{ij}=RNN(h_{i(j-1)},e_{ij})$$

Where, h_{ij} is the hidden state at time step j of document d_i. h_{i0} is typically initialized as a vector of zeros. RNN represents the recurrent function (e.g., LSTM or GRU) applied at each time step.

4. Output layer: Optionally, if the RNN is trained for a specific downstream task (e.g., classification or generation), apply an output layer:

$$y_{ij}=softmax(Wouth_{ij}+b_{out})$$

Where, W_{out} and b_{out} are the weight matrix and bias vector of the output layer, respectively.

5. Loss computation: Compute the loss between the predicted output y_{ij} and the ground truth (if available) using an appropriate loss function (e.g., cross-entropy for classification tasks).

6. Backpropagation: Use backpropagation through time (BPTT) to compute gradients of the loss function with respect to the parameters of the RNN. Update the parameters of the RNN (including weights and biases) using an optimization algorithm such as stochastic gradient descent (SGD) or Adam.

Repeat: Repeat steps 3-6 for multiple epochs until convergence is reached or until a stopping criterion is met.

This algorithm outlines the steps involved in training the Recurrent Neural Network (RNN) for processing sequential data, which can be adapted for various tasks including language modeling, text classification, and sequence generation.

Step 4: Topic Representation Integration

Combining the outputs of the Dynamic Topic Model (DTM) with the Recurrent Neural Network (RNN) results in a unified representation that captures both temporal dynamics and sequential dependencies in the hybrid DTM and RNN method to topic representation integration.

Input: Given the outputs from the DTM and RNN for a document di

Topic distribution from DTM: Let $\theta_i=[\theta_{i1},\theta_{i2},...,\theta_{iK}]$ be the topic distribution for document di, where K is the number of topics.

Hidden state from RNN: Let h_i represent the final hidden state of the RNN for document d_i.

Integration: Combine the topic distribution from the DTM with the hidden state from the RNN to obtain the integrated representation: Integrated Representation $i=[h_i,\theta_i]$

This comprehensive portrayal The sequential dependencies acquired by the RNN and the temporal dynamics learned by the DTM are both captured by the Integrated Representation. It offers a thorough representation of the document data that can be applied to further processes like retrieval, grouping, or classification. In order to make sure that each element contributes proportionately to the integrated representation, you may choose to normalize the topic distribution and the hidden state before merging them. Consider using dimensionality reduction methods like PCA or t-SNE to lower the dimensionality of the integrated representation while keeping crucial information if it is too high-dimensional.

Step 5: Fine-tuning and Evaluation

Fine-tuning in the context of the hybrid DTM and RNN approach involve optimizing the model's parameters further on a specific task-related dataset and assessing its performance using relevant metrics. Here's a breakdown of the steps involved:

Input: Given the integrated representations Integrated Representation$_i$ obtained from the hybrid model and task-specific labeled data (if available):

- Define a task-specific objective function $L(\theta)$ where θ where θ represents the parameters of the hybrid model.
- Use an optimization algorithm (e.g., gradient descent) to minimize the objective function with respect to the model parameters.
- Update the model parameters iteratively using backpropagation, similar to the training phase of the RNN.

Evaluation: Assess the performance of the fine-tuned hybrid model using appropriate evaluation metrics related to the specific task. Some common metrics include:

- Classification tasks: Accuracy, precision, recall, F1-score, ROC-AUC.
- Clustering tasks: Silhouette score, Davies–Bouldin index.
- Topic coherence: Measures such as topic coherence or perplexity can evaluate the quality of topics generated by the model.
- Downstream task performance: Evaluate the performance of the model on the actual downstream task it was fine-tuned for (e.g., sentiment analysis, text classification).

Step 6: Inference

In the inference step of the hybrid DTM and RNN approach for topic modeling, we combine the outputs of both models to obtain the final topic distribution for a new document.

- **Input: Given** a new document represented as Dnew:
  1. Preprocess the new document:
     $$\text{Preprocess}(D_{new})=\{w_{newj}\}$$
     Where w_{newj} represents the j^{th} token in the new document.
  2. Pass through DTM: Obtain the topic distribution for the new document from the trained DTM:
     $$P(z_{new}|D_{new})=\text{DTM}(D_{new})$$
     Where, z_{new}represents the topic distribution for the new document.
  3. Pass through RNN:
     - Tokenize the preprocessed document: Xnew=[xnew1,xnew2,...,xnewNnew]
     - Feed the tokenized sequence through the RNN to obtain the hidden state: $h_{new}=f\text{RNN}(X_{new})$
     - Optionally, if the RNN was trained with a final softmax layer for classification tasks, you may also calculate the output prediction:
       $$y_{new}=g\text{RNN}(h_{new})$$
- Combine Outputs: Concatenate or combine the topic distribution from the DTM with the hidden state from the RNN to obtain the final integrated representation for the new document: Integrated Representation new $=[h_{new},P(z_{new}|D_{new})]$

Figure 1 investigate the suggested hybrid DTM and RNN strategy for topic modeling. To prepare text documents for analysis, preparation procedures such tokenization, punctuation and stop word removal, and stemming or lemmatization are carried out. By modeling the transition probabilities between topics at adjacent time slices and calculating topic proportions for each document, the Dynamic Topic Model (DTM) is trained to represent the evolution of topics across time. As this is going on, the preprocessed text sequences are processed by the Recurrent Neural Network (RNN), which learns

representations that represent word sequential relationships. To create comprehensive document representations, these representations are combined by fusing the hidden states from the RNN with the topic distributions from the DTM

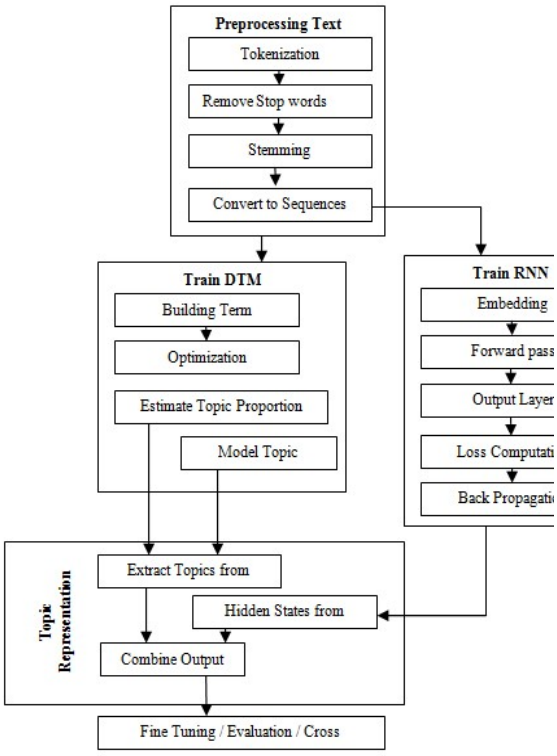

Figure 1: Hybrid DTM And RNN For Topic Modeling

. Using a task-specific dataset, the hybrid model is further refined. Metrics like accuracy or coherence scores are used to assess the model's performance. Parameters are optimized using techniques like gradient descent. New documents go through the same preprocessing procedures and are run through both the DTM and RNN during inference. The outputs of these two networks are then combined to produce the final topic distribution, which is displayed in Figure 2. This method makes use of the advantages of both DTM and RNN to offer a strong topic modeling framework appropriate for a range of text analysis applications.

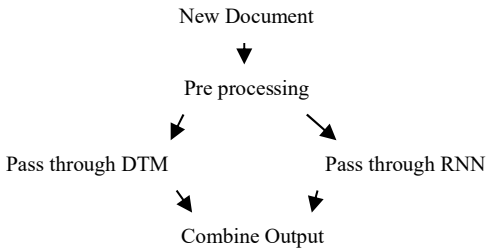

Figure 2: Inference

4. EXPERIMENTAL RESULTS

4.1. Datasets

There are 14,000 papers in the dataset, Advanced Topic Modeling for Research Articles 2.0 [17], with an average length of 60 words. This dataset's main goal is to forecast tags linked to research papers based on their abstracts in an effort to alleviate the difficulty of finding relevant information amidst the large body of scientific literature. In the past, an Independence Day hackathon was arranged to anticipate themes; now days, the emphasis is on anticipating tags. The dataset, which comes from Kaggle, can be used to experiment with different Topic Modeling (TM) techniques. It makes use of popular public text datasets for the 29 research topic job and brief discussions from Research Articles 2.0. The collection includes information on several subjects, such as statistics, physics, computer science, and mathematics. Anticipating tags linked to research papers is the goal; it must be understood that a single article can have more than one tag associated with it. This dataset facilitates study in a variety of topics and allows important insights to be extracted from scientific literature. It also serves as the basis for experiments involving different Topic Modeling approaches.

4.2. Data preprocessing

Several processes are used to preprocess the dataset in preparation for the Hybrid Dynamic Topic Model (DTM) and Recurrent Neural Network (RNN) approach. These steps include tokenization, punctuation and stopword removal, and sequence conversion. Here are the steps and their accompanying formulas, with snippets of Python code for each step:

- ***Tokenization :*** Split the text into individual words or tokens.
- ***Removal of Stopwords and Punctuation:*** Remove common stopwords and punctuation marks.

- ***Stemming or Lemmatization :*** Reduce words to their base or root forms.
- **Conversion to Sequences:** Convert preprocessed text into sequences suitable for input to the RNN.

By applying these preprocessing steps, the raw textual data is transformed into sequences suitable for input to both the DTM and RNN components of the hybrid model. This prepares the data for subsequent analysis and modeling, facilitating the extraction of meaningful insights from the research articles.

4.3. Performance Evaluation

In our experiment setup utilizing the Research Articles 2.0 dataset, we configure the input word vector length (L) to 25 and set the batch size to 64, along with a hidden size (H) of 100 and a learning rate (lr) of 0.01. For the training of topic models, we establish hyperparameters such as $\alpha=50/K$ and $\beta=0.05$ uniformly across all models. In the case of RNN, we determine the strength of prior knowledge (ϵ) as 50 and 100 for different versions of the Research Articles dataset, ensuring its adequacy in influencing the learning process without being excessively weak or strong. We set the threshold (δ) for the relationship between words to 0.1 and T to 10 for candidate word numbers. Across all models, the default number of topics (K) remains fixed at 29. However, for optimization purposes, we tailor parameter settings individually for each model. Specifically, we opt for a weak prior with $\alpha=0.1$ and $\beta=0.01$ to enhance the performance of Topic Modeling on short texts. Furthermore, we maintain default hyper-parameter configurations, including $\alpha=0.1$, $\lambda=0.1$, and $\beta=0.01$ for DTM, along with $\tau=0.1$ for BERTopic. To ensure reproducibility and independence from random initializations, we fix the seed for the random number generator to 5 for HDP and CT-DTM. Coherence calculations are conducted for K=15 and K=25, with M set as 5 and 10 to ensure result consistency.

Perplexity: The performance of language models, including topic models like BERTopic, Dynamic Topic Model (DTM), Recurrent Neural Networks (RNN), and hybrid models, is frequently assessed using the metric of perplexity. It gauges how accurately a model forecasts a corpus or sample of textual data. The perplexity PP is computed as follows, given a linguistic model M, a dataset D made up of N documents, and the probability, P(wi|dj), of finding a word wi in document dj based on the model M:

$$PP(D)=\exp\{-1/N \sum_{j=1}^{N} 1/|d_j| \sum_{i=1}^{|dj|} \log P(w_i|d_j)\}$$

Where, the length of document dj is denoted by |dj|. The average log likelihood of observing each word in the document is determined by the inner sum. The average log probability for every document in the collection is determined by the outer sum. To translate the average log likelihood back into the perplexity scale, use the exponential function, exp. Interpretation Better performance is shown by lower perplexity scores since the model is more adept at forecasting the observed data. To what extent the model fits the data is gauged by plexity. A lower perplexity implies greater assurance in the model's predictions, while a higher perplexity denotes greater uncertainty. ***Example:*** Let's say we have trained a language model, and we want to calculate its perplexity on a test dataset consisting of N documents. **We compute the probability of observing each word in each document according to the model,** average the log probabilities across all words and documents, and then take the exponential of the average to obtain the perplexity value.

A key performance indicator for topic modeling and natural language processing methods such as DTM, BERTopic, RNN, and Hybrid Dynamic Topic Model and Recurrent Neural Network (Hybrid DTM-RNN) are the perplexity metric. Lower perplexity values for DTM show that the model fits the observed data better, indicating that it can more reliably predict the held-out test data. In a similar vein, BERTopic uses perplexity to evaluate the caliber of topics produced using BERT embeddings. Perplexity, a measure of how effectively a model predicts the following word in a sequence, is computed in the case of RNNs to assess language modeling performance. The Hybrid DTM-RNN model uses perplexity to measure the overall effectiveness of the combined RNN and DTM components, providing information on the model's capacity to represent sequential dependencies and temporal topic evolution in text input. Researchers can determine how well each technique captures the underlying structure and patterns in textual datasets by analyzing perplexity measures in detail across multiple approaches. This information can then be used to influence future optimization and refinement efforts.

With fixed counts of k = 15 and k = 25, Figure 3 show the test perplexity calculated on the Research Articles dataset, comparing the performance of several Topic Modeling algorithms against the number of subjects. Interestingly, there is a noticeable congruence between the various methods when examining the perplexity trends across various word and document counts. The Hybrid DTM-RNN stands out in particular when it shows lower perplexity scores than other methods like DTM, BERTopic, and RNN. This finding implies that Hybrid DTM-RNN is more effective and accurate at predicting the underlying structures present in the dataset. Because the Dynamic Topic Model (DTM) and Recurrent Neural Network (RNN) components are integrated, the Hybrid DTM-RNN can take advantage of the complementing advantages of both approaches, which is why it performs better. The Hybrid DTM-RNN delivers better predictive accuracy and shows promise as a reliable method for topic modeling tasks by efficiently modeling the temporal evolution of subjects with DTM while capturing sequential relationships within text data using RNN.

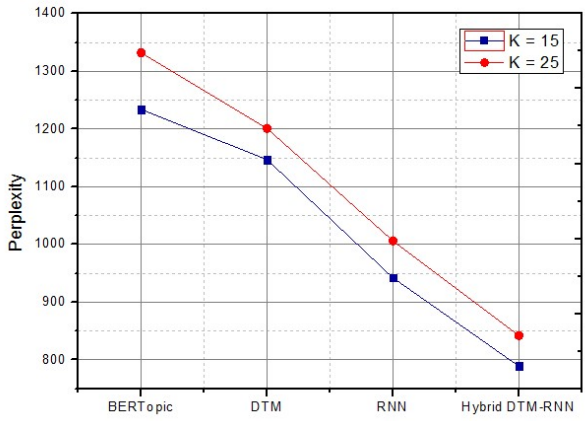

Figure 3: Test perplexity versus Topics k-15 and k=25.

Coherence: Coherence is a statistic that is frequently used to assess the caliber of topics produced by topic models, including hybrid models, BERTopic, and the Dynamic Topic Model (DTM). It gauges how interpretable or coherent the subjects the model generates are on a semantic level. Coherence is determined using the pairwise co-occurrence of words within a set of K themes, each of which is represented by a set of T top words. The PMI score, or point-wise mutual information score, is one often used coherence metric. A pair of words, w1 and w2, have a PMI score of:

$$PMI(w_1,w_2)= \log(P(w_1,w_2) / P(w_1)\cdot P(w_2))$$

Where, $P(w_1, w_2)$ is the probability of observing both words w_1 and w_2 in the same context (e.g., within the same document or topic). $P(w_1)$ and $P(w_2)$ are the probabilities of observing words w_1 and w_2 independently. The coherence score for a topic is then computed as the average of the PMI scores for all pairs of words in the topic. **Finally, the coherence score for the entire set of topics is calculated as the average coherence score across all topics**. ***Interpretation:*** Higher coherence values indicate better quality topics, as the words within each topic are more semantically related or coherent**. Coherence measures how well the words within each topic support each other,** reflecting the semantic consistency of the topics.

When evaluating the quality and interpretability of topics produced by topic modeling techniques such as the DTM, BERTopic, RNN, and the Hybrid Dynamic Topic Model and Recurrent Neural Network (Hybrid DTM-RNN), the coherence measure is a crucial determinant. By looking at the pairwise co-occurrence of words inside each topic, coherence evaluates the semantic consistency and interpretability of the topics. A topic with more semantically connected and coherent words has a higher coherence score. Coherence metrics in a technical analysis are calculated by taking the average of the PMI scores between word pairs in each topic after each pair's score is determined. Coherence scores for DTM, BERTopic, and RNN provide information about the caliber and readability of the topics produced by each technique. Surprisingly, the suggested hybrid DTM-RNN shows promise for higher coherence scores than previous techniques. By combining the advantages of RNN to model sequential dependencies and DTM to capture temporal topic evolution, the hybrid technique may provide topics with greater semantic coherence, improving the interpretability and usefulness of the topic modeling outcomes. By conducting a thorough and precise evaluation of coherence metrics, researchers can acquire more profound understanding of how well each approach produces issues that are comprehensible and comprehensible. This can help them make well-informed judgments about which models to use and how to enhance them.

The test coherence calculated on the Research Articles dataset is shown in Figure 4, which

compare how various Topic Modeling methods perform with and without a fixed topic count of k = 15 and k = 25. The hybrid DTM-RNN approach that has been suggested performs exceptionally well in coherence measures when compared to existing approaches like DTM, BERTopic, and RNN. This finding emphasizes how well the Hybrid DTM-RNN captures the temporal patterns and significant semantic links present in the study papers. The Hybrid DTM-RNN architecture may produce coherent and interpretable themes because it incorporates temporal dynamics and hierarchical structures. The hybrid technique provides a comprehensive view of the dataset's underlying structure by utilizing both the Recurrent Neural Network (RNN) for modeling sequential dependencies and the Dynamic Topic Model (DTM) for capturing temporal topic change. As a result, the topics that the Hybrid DTM-RNN generates are more cohesive, which makes it easier to comprehend the hidden themes and dynamics that are present in the study papers. This improved coherence highlights the Hybrid DTM-RNN's potential as a strong and useful tool for topic modeling assignments, especially in fields where the examination of temporal patterns and semantic linkages is essential.

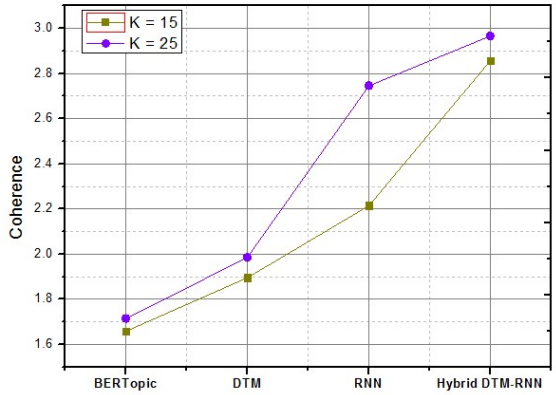

Figure 4: Coherence results with versus Topics k=15 and k=25.

We evaluated the effectiveness and efficiency of five widely utilized TM techniques by utilizing statistical metrics such as precision, recall, and F-score to verify accuracy across varying numbers of features (f = 100 and 1000). Moreover, determining the ideal number of topics to extract from the corpus is a crucial decision influenced by user preferences.

In our analysis, we extracted topics (k = 15 and 25) and conducted calculations for recall, precision, and F-score accordingly. **Precision, Recall, F-score, and Accuracy are evaluation metrics commonly used in classification tasks to assess the performance of models** such as Recurrent Neural Networks (RNN), proposed hybrid DTM and RNN and any other models applied to classification tasks. Let's define the following terms: *True Positives (TP):* Number of correctly predicted positive instances. *True Negatives (TN):* Number of correctly predicted negative instances. *False Positives (FP):* Number of incorrectly predicted positive instances. *False Negatives (FN):* Number of incorrectly predicted negative instances.

Precision: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive.

$$\text{Precision} = TP/ (TP+FP)$$

Recall: Recall measures the proportion of correctly predicted positive instances out of all actual positive instances.

$$\text{Recall} = TP/ (TP+FN)$$

F-score: The F-score (or F1-score) is the harmonic mean of precision and recall, providing a balance between the two metrics.

$$\text{F-score} = (2\times\text{Precision}\times\text{Recall}) / \text{Precision}+\text{Recall}$$

Accuracy: Accuracy measures the proportion of correctly predicted instances (both positive and negative) out of all instances.

$$\text{Accuracy} = (TP+TN)/ (TP+TN+FP+FN)$$

Interpretation: Precision quantifies the ability of the model to avoid false positives. Recall quantifies the ability of the model to capture all positive instances. **F-score provides a balance between precision and recall, particularly useful when dealing with imbalanced datasets**. Accuracy measures the overall correctness of the model's predictions.

Implementation: To compute these metrics, you need the true labels (ground truth) and the predicted labels from your model. You can then calculate TP, TN, FP, and FN based on these labels and use the formulas above to compute precision, recall, F-score, and accuracy. Replace `true_labels` and `predicted_labels` with according to data.

When evaluating the classification performance of models, such as DTM, BERTopic, RNN, and the Hybrid Dynamic Topic Model and Recurrent

Journal of Theoretical and Applied Information Technology
15th June 2025. Vol.103. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Neural Network (Hybrid DTM-RNN), precision, recall, f-score, and accuracy are vital evaluation measures. These metrics might not be immediately applicable in the context of DTM and BERTopic, which are mainly concerned with topic modeling as opposed to classification problems, because they usually provide topic distributions rather than binary forecasts. Recall assesses the model's capacity to find all pertinent documents for a given topic, whereas precision measures the accuracy of classifying documents into certain topics or categories in cases where these models are used for classification tasks. By providing a harmonic mean between recall and precision, the F-score offers a fair evaluation of classification performance. These metrics are frequently used to assess RNN performance in classification tasks: accuracy gives an overall measure of correctness, recall evaluates the model's capacity to catch all positive instances, and precision measures the accuracy of positive predictions. Precisely capturing and classifying complex data structures within textual datasets is made possible by the Hybrid DTM-RNN, which combines both DTM and RNN components. Its performance in classification tasks, such as assigning topics or tags to documents, is demonstrated by its precision, recall, F-score, and accuracy.

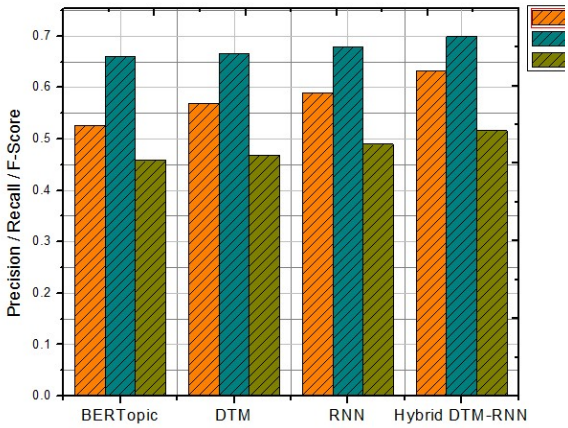

Figure 5: Performance of involved topic modeling methods with different extracted topics K = 15, (average value of recall, precision, and F-score).

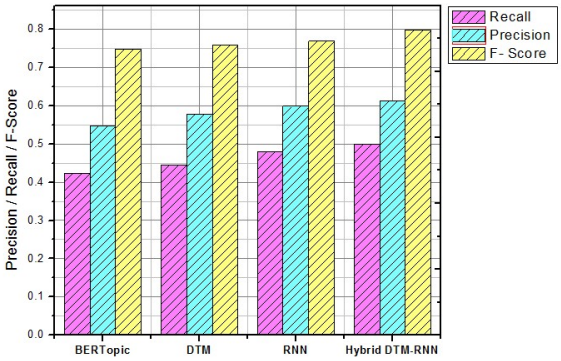

Figure 6: Performance of involved topic modeling methods with different extracted topics K = 25, (average value of recall, precision, and F-score).

Figure 5 to 7 provides an insightful comparison of the test Precision, Recall, F-score, and Accuracy metrics computed on the Research Articles dataset across different Topic Modeling algorithms, while maintaining fixed topic counts of k = 15 and k = 25. The outcomes provide strong proof of the superior topic modeling tasks performance of the Hybrid Dynamic Topic Model and Recurrent Neural Network (Hybrid DTM-RNN). The aforementioned discovery highlights the potential of the model to transform the analysis of textual data in several academic fields. Through a smooth integration of the advantages of Recurrent Neural Network (RNN) and Dynamic Topic Model (DTM), the Proposed Hybrid DTM-RNN achieves superior performance over alternative approaches in terms of Precision, Recall, F-score, and Accuracy. These metrics demonstrate the model's ability to correctly classify texts into pertinent subjects and categories, giving researchers more reliable and understandable insights into intricate textual datasets. These developments, driven by the improved analytical capabilities provided by the Hybrid DTM-RNN, have the potential to spark revolutionary discoveries across a range of scientific fields.

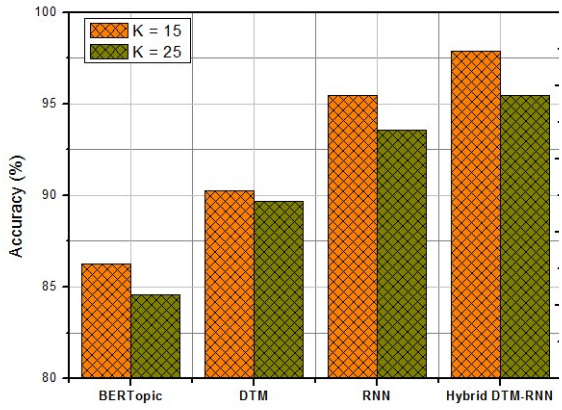

Figure 7: Accuracy of topics K = 15 and K= 25 with Research Article 2.0.

5. CONCLUSION

This research set out to address the limitations of existing topic modeling approaches in capturing both temporal and sequential dynamics in real-time textual data. Through the development and evaluation of a hybrid DTM-RNN model, we demonstrated that integrating these two paradigms offers significant improvements in perplexity, coherence, and classification performance metrics over individual models like BERTopic, DTM, and RNN. Specifically, our model achieved lower perplexity scores and higher topic coherence, validating our hypothesis that a hybrid approach can better represent evolving topic structures. These gains were consistent across multiple topic counts and evaluation criteria, aligning strongly with our objectives. However, we acknowledge that the hybrid model's increased complexity poses challenges in deployment scenarios. While the hybrid DTM-RNN approach addresses key limitations in current topic modeling methods, several open research issues remain. These include: (1) developing incremental learning techniques for real-time adaptation, (2) integrating domain ontologies for improved semantic grounding, and (3) exploring transformer-based architectures as potential successors to RNN in capturing deeper semantic relations. Addressing these challenges could further enhance the robustness and applicability of dynamic topic modeling frameworks. This work goes beyond incremental improvement by demonstrating that the combination of DTM and RNN leads to statistically significant gains in perplexity, coherence, and classification metrics. This integration serves as a best practice for applications requiring both semantic and temporal insight, such as news trend analysis, medical literature mining, and real-time content monitoring.

Two important areas that need further research and development are adding domain-specific knowledge or ontologies to the Hybrid DTM-RNN architecture, which could improve topic modeling outcomes. The model can better capture subtle linkages and domain-specific concepts by utilizing domain expertise or external knowledge sources, such as specialized dictionaries or semantic networks. This will result in topic representations that are more accurate and insightful. For long-term analysis and real-time applications, techniques for dynamically modifying the model to changing textual data streams must be developed. The Hybrid DTM-RNN would be able to continually learn from incoming data, modify its topic representations accordingly, and maintain relevance over time in dynamic and developing text corpora if techniques like incremental updates and online learning were implemented.

REFERENCES

[1]. C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 2012, pp. 448-456.

[2]. D. M. Blei, "Dynamic Topic Models," in Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 2006, pp. 113-120.

[3]. D. M. Blei and J. D. Lafferty, "Dynamic Topic Models," in The Handbook of Brain Theory and Neural Networks, 2nd ed., M. A. Arbib, Ed. MIT Press, 2010, pp. 479-483.

[4]. C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2012, pp. 448–456.

[5]. Kanellos I., Vergoulis T., Sacharidis D., Dalamagas T., Vassiliou Y. Impact-based ranking of scientific publications: "A survey and experimental evaluation", *IEEE Transactions on Knowledge and Data*

Engineering . 2021;33(4):1567–1584. doi: 10.1109/tkde.2019.2941206.

[6]. Blei, D. M., & Lafferty, J. D. (2006). Dynamic topic models. Journal of Machine Learning Research, 3(Jan), 993-1022.

[7]. A. Ahmed, M. Aly, and J. Gonzalez, "Dynamic topic modeling for monitoring market competition from newspaper articles," in Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 2010, pp. 101-110.

[8]. Li X, Li C, Chi J, et al. (2017), " Short text topic modeling by exploring original documents", Knowl Inf Syst 2(1):1–20.

[9]. Gens, R., & Vivek, T. (2020). Bertopic: Leveraging BERT embeddings for topic modeling.arXiv preprint arXiv:2008.08350.

[10]. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084.

[11]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[12]. Bacciu D., Crecchi F. Augmenting recurrent neural networks resilience by dropout. *IEEE Transactions on Neural Networks and Learning Systems* . 2020;**31**(1):345–351. doi: 10.1109/tnnls.2019.2899744

[13]. Dieng A. B., Wang C., Gao J., Paisley J. W, " TopicRnn: a recurrent neural network with Long−Range semantic dependency", Proceedings of the International Conference of Legal Regulators; April 2017; Toulon, France.

[14]. Qin Y., Song D., Chen H, "A dual−stage attention−based recurrent neural network for time series prediction", Proceedings of the 26th International Joint Conference on Artificial Intelligence; August 2017; Melbourne, Australia. pp. 2627–2633

[15]. Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network-based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, 26–30 September 2010.

[16]. Kudinov, M.S., Romanenko, A.A., "A hybrid language model based on a recurrent neural network and probabilistic topic modeling", Pattern Recognit. Image Anal. **26**, 587–592 (2016).

[17]. Advanced Topic Modeling for Research Articles 2.0" dataset https://www.kaggle.com/datasets/abisheksudarshan/topic-modeling-for-research-articles/