GMR

The Original Semantic-Aware Topic Modeling in Medical Texts Using A DTM-RNNLSTM Framework with UMLS Integration

S. Jayabharathi [1], Dr. M. Logambal [2]

[1] *Research Scholar, Department of Computer Science, Vellalar College for Women (Autonomous), Thindal, Erode, Tamil Nadu, India. E-Mail ID: jayabharathi8383@gmail.com*
[1]*Assistant Professor, Department of Computer Applications, K. S. Rangasamy College of Arts & Science (Autonomous), Tiruchengode, Namakkal, Tamil Nadu, India. E-Mail ID: jayabharathi8383@gmail.com*
[2] *Associate Professor, Department of Computer Science, Vellalar College for Women (Autonomous), Thindal, Erode, Tamil Nadu, India. E-Mail ID: m.logambal@vcw.ac.in*

ABSTRACT:
The exponential rise in unstructured medical text volume in recent years has led to a pressing need for sophisticated topic modeling methods that can capture temporal dynamics and semantic richness. This study suggests a brand-new hybrid framework called DTM-RNNLSTM, which combines the sequential learning powers of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks with Dynamic Topic Modeling (DTM). The model integrates ideas from the Unified Medical Language System (UMLS) to improve semantic relevance, making it possible to identify issues with medical significance. The MedMentions dataset, a sizable corpus annotated with UMLS concepts, is used to assess the efficacy of the suggested model. Three robust baseline models are compared: the Dynamic Topic Model (DTM), Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), and Non-negative Matrix Factorization (NMF). Coherence, Perplexity, Precision, Recall, F1-Score, and Accuracy are evaluation measures that address both statistical and semantic performance factors. The findings show that DTM-RNNLSTM outperforms conventional methods in capturing changing topic patterns and greatly enhances semantic coherence.

Keywords: *Coherence and Perplexity, Dynamic Topic Model (DTM), Long Short-Term Memory (LSTM), Medical Natural Language Processing (NLP), MedMentions Dataset, Recurrent Neural Networks (RNN), Semantic Integration, Topic Modeling, Unified Medical Language System (UMLS).*

INTRODUCTION

The amount of textual data being generated in the medical industry is increasing at an unprecedented rate. A large and continuously expanding collection of unstructured medical texts is facilitated by Electronic Health Records (EHRs), clinical trial reports, scientific research papers, and online health forums [1]. The digital transformation of healthcare systems and the growing focus on data-driven medical research have further accelerated this surge. Advanced natural language processing (NLP) techniques are crucial because of the unstructured nature of this data, which makes it difficult to extract insights that may be put to use [2]. In order to organize, summarize, and uncover hidden patterns in massive text corpora, topic modeling has become an essential tool. Topic models help with a number of medical applications, including public health monitoring, clinical decision assistance, illness trend analysis, and literature review automation [3]. These models improve knowledge discovery, information retrieval, and evidence-based medical research by offering a probabilistic framework for identifying hidden themes in text data.

Traditional topic modeling techniques, such Latent Dirichlet Allocation (LDA) [4], Non-negative Matrix Factorization (NMF) [5], and even Dynamic Topic Models (DTM) [6], are useful, but they face two major obstacles in the medical field: temporal dynamics and semantic understanding. First, because these models are mainly statistical in nature, they frequently ignore the domain-specific semantics present in biological texts, producing subjects that are ambiguous or clinically irrelevant. Second, DTM does not adequately simulate long-term dependencies or context preservation across changing document sequences, even while it does capture temporal changes in subjects. Furthermore, these approaches overlook outside medical expertise, such the Unified Medical Language System (UMLS) [7], which could greatly improve the topic's relevance and interpretability. This highlights the necessity for a hybrid modeling approach that integrates deep sequential learning, domain-specific semantic enrichment, and temporal awareness. The U.S. National Library of Medicine created the Unified Medical Language System (UMLS), a comprehensive biological vocabulary collection that incorporates more than 200 medical classifications and terminologies. It enables uniform understanding of medical language by offering a standardized mapping of concepts and synonymous phrases across several healthcare areas. Because it facilitates entity linkage, concept disambiguation, and semantic

normalization, UMLS is very useful for natural language processing (NLP) applications. A key component of precise topic modeling in the healthcare industry, UMLS improves the semantic interpretability of medical data by bringing unstructured text into line with structured medical ideas [8].

Although current topic modeling techniques have shown useful in identifying themes in medical corpora, they frequently fall short in their ability to accurately represent temporal transitions and integrate semantic understanding. Conventional models cannot capture changing topics in longitudinal medical datasets well, and they are not able to leverage domain-specific information such as UMLS. Furthermore, the sequential character of medical narratives, such patient histories or time-stamped articles, is difficult for current techniques to depict. This disparity limits the themes' applicability in clinical or research settings by impeding the development of cogent and medically significant topics.

This study suggests a sophisticated hybrid model and an all-encompassing assessment approach to improve subject modeling in medical literature in order to overcome these issues. Dynamic Topic Models (DTM) [9] and the sequential learning power of Recurrent Neural Networks (RNN) [10] and Long Short-Term Memory (LSTM) [11] networks are combined in this innovative framework. Through this fusion, the model is able to preserve context and long-range dependencies in the text while capturing the temporal evolution of the topic. The model's semantic awareness is improved with the use of UMLS concept annotations during preprocessing. This enables it to produce themes that are interpretable and medically relevant, matching clinical concepts and terminology from the real world. Three well-known topic modeling approaches are used to compare the suggested model: Dynamic Topic Model (DTM) [13], Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), and Non-negative Matrix Factorization (NMF) [12]. Metrics like Coherence, Perplexity, Precision, Recall, F1-Score, and Accuracy can be used to assess the statistical and semantic performance of the suggested approach thanks to these baselines. By integrating deep learning, temporal modeling, and semantic integration, this study offers a fresh approach to medical topic modeling with the goal of improving the caliber and usefulness of knowledge extraction in biomedical text mining.

This paper's remaining sections are arranged as follows: The related work is covered in Section II, with an emphasis on current methods for temporal sequence learning and dynamic topic modeling in clinical or medical text analysis. A Dynamic Topic Model (DTM) layer, a Recurrent Neural Network/Long Short-Term Memory (RNN/LSTM) layer for capturing temporal coherence, and a fusion layer intended to maximize topic coherence are all integrated in the suggested approach, DTM-RNNLSTM, which is described in Section III. The application of UMLS-based preprocessing for precise clinical concept annotation is also covered in this section. The data preprocessing procedures used to get the input ready for model training are described in Section IV. The setup and thorough performance analysis utilizing coherence scores, perplexity, and common classification metrics like accuracy, precision, recall, and F1-score are presented in Section V along with the experimental findings and discussion. The work is finally concluded in Section VI, which also suggests future research areas.

RELATED WORK

For the purpose of organizing and comprehending vast amounts of unstructured biomedical text, topic modeling has become a crucial tool. Conventional models like Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) [14] and Non-negative Matrix Factorization (NMF) (Lee & Seung, 2001) [15] have been used extensively in biomedical literature to uncover latent semantic structures, but they frequently fail to capture the particular nuances of clinical and biomedical language, such as synonymy and domain-specific terminology. To overcome the shortcomings of static models, Dynamic Topic Models (DTM) were introduced to capture topic evolution over time (Blei & Lafferty, 2006) [16]. DTM has been used in the healthcare industry to research topics, disease prevalence, and patient record evolution.

DTM is useful for simulating temporal changes, but it is unable to identify the more intricate sequential patterns and contextual connections found in longitudinal data. In order to improve text representation, recent developments in deep learning have brought models such as Neural Variational Document Models (NVDM) and hybrid models that combine Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (Miao, Yu, & Blunsom, 2016) [17]. These models do better than traditional methods at capturing document-level dependencies and context. Due to a lack of semantic interaction with outside medical information sources, their use in medical subject modeling is still restricted. By connecting unstructured text to organized biomedical concepts, the Unified Medical Language System (UMLS) has been utilized to improve medical natural language processing applications. Research has demonstrated that UMLS-based semantic annotation improves text categorization, entity recognition, and concept normalization (Limsopatham & Collier, 2016; Wang et al., 2018). [18] [19]. UMLS incorporation into subject modeling frameworks remains understudied despite its demonstrated advantages. Proposed by Yin and Wang (2014) [20], the Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) model has proven to perform better when processing brief texts like clinical notes or medical papers. GSDMM is useful for grouping brief segments together, but it is unable to incorporate domain-specific semantics or represent temporal evolution. Each of the current subject modeling approaches has special advantages. RNN/LSTM for sequential learning, UMLS for semantic enrichment, and DTM for temporal modeling. To satisfy the requirements of dynamic, semantically complex medical datasets, no existing model effectively integrates these features. By putting up a hybrid DTM-RNNLSTM architecture combined with UMLS for improved topic modeling in biomedical texts, this study seeks to close this gap.

Proposed Methodology: DTM-RNNLSTM

The hybrid architecture of the suggested DTM-RNNLSTM model was created to get beyond the drawbacks of conventional topic modeling in medical texts. This is accomplished by combining the context-preserving and sequence-modeling power of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) units with the temporal topic evolution capabilities of Dynamic Topic Models (DTM). Through UMLS-based preprocessing, it also improves semantic understanding, allowing the model to provide themes with clinical significance. The following elements make up the DTM-RNNLSTM architecture:

Dynamic Topic Model (DTM) Layer

Dynamic Topic Models (DTMs), extend Latent Dirichlet Allocation (LDA) to capture temporal topic evolution. In this model, the topics are allowed to vary over discrete time slices, making it suitable for modeling corpora where content changes over time—such as medical literature or clinical notes. DTM is employed as the first component to model topic evolution over time. The input corpus is split into time-stamped segments (e.g., by publication year or patient admission date). For each time slice, DTM learns a set of latent topics and their distribution across documents, capturing how these topics evolve temporally.

Step 1: Temporal Segmentation of the Corpus

Let the entire corpus D be divided into T time slices: $D = \{D^{(1)}, D^{(2)}, \ldots, D^{(T)}\}$

Each $D^{(t)}$ represents a sub-corpus of documents corresponding to time slice t (e.g., a publication year).

Step 2: Generative Process of DTM

For each time slice $t \in \{1,\ldots,T\}$:

1. For each topic $k \in \{1,\ldots,K\}$:
   - The topic-word distribution $\beta_k^{(t)}$ is drawn from a Gaussian random walk in the natural parameter space (log-space):

$$\eta_k^{(t)} \sim N\left(\eta_k^{(t-1)}, \sigma^2 I\right), for\ t > 1$$

$$\beta_k^{(t)} = softmax(\eta_k^{(t)})$$

   - This allows each topic's word distribution to evolve smoothly over time.
2. For each document $d \in D^{(t)}$:
   - Draw document-topic distribution: $\theta_d \sim Dir(\alpha)$
   - For each word w_{dn}wdn in document d:
     - Draw a topic assignment: $z_{dn} \sim Multinomial(\theta_d)$
     - Draw a word: $w_{dn} \sim Multinomial(\beta_{z_{dn}}^{(t)})$

Where, $\eta_k^{(t)}$- Natural parameter for topic k at time t (logit of β), $\beta_k^{(t)}$- Word distribution for topic k at time t, θ_d -Topic distribution for document d, Z_{dn} -Topic assignment for the n-th word in document d. w_{dn} - The n-th word in document d.

Inference and Learning

DTM uses Variational Inference or Kalman Filtering with Expectation-Maximization (EM) to estimate:

- The topic trajectories $\{\eta_k^{(t)}\}_{k=1}^K$
- Document-topic distributions θ_d
- Per-word topic assignments z_{dn}

In some implementations, Variational Kalman Filtering is employed for efficient time series inference. The overall goal is to maximize the Evidence Lower Bound (ELBO) across time slices:

$$L = \sum_{t=1}^{T} E_{q(\theta,z)}[log\ log\ P(D^{(t)}|\theta, z, \beta^{(t)})] - KL[q(\theta, z)||P(\theta, z)]$$

Where, $P(\beta^{(t)}|\beta^{(t-1)}$acts as a temporal prior, KL is the Kullback–Leibler divergence between variational and true posteriors. This process ensures that topic distributions are temporally coherent, with smooth transitions between adjacent time slices—making DTM a powerful foundation for sequential modeling with RNN-LSTM layers.

RNN/LSTM Layer for Temporal Coherence
The purpose of integrating the RNN/LSTM layer with the DTM is to model long-term dependencies and contextual transitions across document-topic distributions over time. This is especially important in medical texts, where topics may shift slowly, be influenced by previous contexts, or appear intermittently.

Step 1: Preparing Input for RNN/LSTM
From the DTM component, it obtains document-topic distributions $\theta_d^{(t)} \in R^K$ for each document d in time slice t, where K is the number of topics.

It aggregates these distributions into time series of topic vectors per document or per aggregated entity (e.g., disease class, concept cluster):

$$\Theta^{(d)} = \left[\theta_d^{(1)}, \theta_d^{(2)}, \ldots, \theta_d^{(T)}\right] \in R^{TxK}$$

This matrix becomes the input sequence for the RNN/LSTM layer.

Step 2: Recurrent Modeling with LSTM
LSTM (Long Short-Term Memory) networks are designed to capture long-range dependencies in sequential data. Unlike vanilla RNNs, LSTMs mitigate the vanishing/exploding gradient problem through their gating mechanisms. Let's define the input to the LSTM at time t as:

$$x_t = \theta_d^{(t)}$$

The LSTM unit computes hidden states $h_t \in R^H$ and cell states $c_t \in R^H$using the following equations:

LSTM works as follow,

Forget Gate: Computes a sigmoid activation over the current input and previous hidden state, outputting values between 0 and 1 to decide which parts of the previous memory cell C_{t-1} should be forgotten.

$$\boldsymbol{f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)}$$

When processing biomedical texts from MedMentions, the forget gate learns to selectively remove outdated or irrelevant medical concepts from memory. For example, if earlier tokens discussed "diabetes" but the context shifts to "cardiovascular disease," the forget gate helps discard diabetes-related memory.

Input Gate: Computes another sigmoid to decide which new values will be updated in the memory cell.

$$\boldsymbol{i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)}$$

This gate controls the addition of new UMLS-based concepts into the model's memory. When a new clinical term like "angioplasty" appears, the input gate decides how strongly this new information should influence the next topic state.

Candidate Memory (Cell Candidate): Computes a tanh-activated vector of new candidate values C_t that could be added to the state.

$$\boldsymbol{C_t = \sigma(W_C.[h_{t-1}, x_t] + b_C)}$$

Generates potential updates to the LSTM's memory, representing possible new biomedical concepts or topic shifts (e.g., proposing a new cluster around "heart conditions" based on current input tokens).

Cell State Update: Updates the cell state C_t by combining the forget gate and input gate results.

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t$$

The forget and input gates combine their outputs to update the internal memory cell. Old, less-relevant concepts are erased, and new clinical concepts are integrated, maintaining an up-to-date semantic understanding.

Output Gate: Another sigmoid activation deciding which part of the updated cell state forms the output.

$$\boldsymbol{o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)}$$

Determines what parts of the updated memory are output to influence the next hidden state. It ensures that only the medically relevant features (e.g., symptoms, procedures) are propagated for subsequent topic prediction.

Hidden State: The final hidden state h_t is derived by applying tanh activation to C_t and gating it by o_t.

$$h_t = o_t \odot tanh\ (C_t)$$

The semantic summary of the document up to that moment is encoded in the hidden state at each time step, which aids the DTM-RNNLSTM model in more accurately predicting the changing medical subjects. where H is the hidden dimension size, W, U, and b are learnable parameters, σ is the sigmoid activation, and ⊙ indicates element-wise multiplication. LSTM predicts logical subject transitions, updates with new clinical knowledge, and dynamically forgets unrelated medical topics in MedMentions. In theory, it models fine-grained semantic evolution in biomedical papers by updating internal memory and gating it at each token step.

Step 3: Output Interpretation

- The final hidden state h_T (or the full sequence $\{h_1,...,h_T\}$ encodes context-aware temporal topic transitions.
- This can be used to:
    - Forecast topic distributions at future time steps
    - Smooth noisy or erratic topic changes
    - Cluster documents based on temporal topic trajectories
    - Provide better classification features for downstream tasks

Loss Function for Sequence Prediction (Optional)
If supervised (e.g., forecasting topic vector θ(T+1), use Mean Squared Error (MSE):

$$L_{MSE} = \frac{1}{K}\sum_{i=1}^{K} \left(\hat{\theta}_i^{(T+1)} - \theta_i^{(T+1)}\right)^2$$

If unsupervised (e.g., smoothing), a reconstruction loss between predicted and actual topic vectors can be used. Benefits of RNN/LSTM Integration are captures temporal dependencies across document-topic distributions. Learns non-linear transitions that traditional DTM may miss. Handles irregular time steps and sparse topic changes, common in medical datasets.

Fusion Layer and Topic Coherence Optimization
This layer integrates the temporal context captured by the LSTM with the original topic distribution outputs from DTM. The idea is to produce semantically enriched**,** temporally coherent topic distributions that can be used to refine topic-word relationships and improve interpretability and performance.

Step 1: Fusion via Softmax Transformation
Let the final hidden state of the LSTM for a document d be:

$$h_d^{(T)} \in R^H$$

To convert this high-dimensional hidden representation back into a topic distribution vector $\widehat{\theta}_d \in R^H$ (where K is the number of topics), apply a fully connected layer followed by softmax**:**

$$\widehat{\theta}_d = softmax(W_h h_d^{(T)} + b_h)$$

Where, $W_d \in R^{k\,x\,t}$ is the learnable weight matrix, $b_d \in R^k$ is the bias vector, $\widehat{\theta}_d$ represents the re-estimated topic distribution with enhanced temporal coherence.

Step 2: Reconstruction of Topic-Word Distributions
Using the temporally smoothed topic distributions $\widehat{\theta}_d$, reconstruct the topic-word matrix $\phi_{k,w}$, which encodes the probability of word w under topic k. A common approach is to use matrix factorization or soft attention over word embeddings:

$$\phi_{k,w} = \frac{exp\ (E_k^T . V_w)}{\sum_{w'} \quad (E_k^T . V_{w'})}$$

Where: $W_k \in R^D$ is the embedding of topic k, $V_w \in R^D$ is the embedding of word w, D is the embedding dimension Alternatively, one may learn $\phi_{k,w}$ directly using:

$$\phi_k = softmax(W_k V + b_k)$$

Step 3: Topic Coherence Optimization Objective
To optimize the semantic coherence of the learned topics, use Topic Coherence Loss, such as Normalized Pointwise Mutual Information (NPMI) or UMass coherence. For UMass coherence (simpler and fast), given top-N words of topic k: $\{w_1,w_2,...,w_N\}$,

$$Coherence_{UMass}(k) = \sum_{i-2}^{N} \sum_{j=1}^{i-1} log \frac{D(w_i, w_j)+\epsilon}{D(w_j)}$$

Where, $D(w_i,w_j)$ is the co-occurrence count of words w_i and w_j in documents, ϵ is a small smoothing constant. This loss can be incorporated into training as:

$$L_{total} = L_{reconstruction} + \lambda . L_{coherence}$$

Where, $L_{reconstruction}$ is the cross-entropy loss between original and predicted topic distributions or topic-word distributions. λ is a weighting factor to control coherence regularization

The suggested strategy has a number of advantages. First, temporal coherence is accomplished by the RNN/LSTM layer's use of hidden states, which gradually smooth out sudden or noisy topic changes to provide more logically consistent topic transitions. Second, the model facilitates semantic refinement, which improves the clarity of topic representation by making the word distributions within subjects more meaningful and interpretable. Finally, by limiting topic creation to highlight UMLS concepts, the approach facilitates UMLS-aware reconstruction, which enables the integration of domain-specific information. Learned themes can be more closely aligned with clinical semantics by using strategies like topic-word masking or weighting algorithms based on UMLS relevance scores.

UMLS-Based Preprocessing for Concept Annotation:
To enrich the input data with biomedical semantics, UMLS-based preprocessing is performed as follows:
Concept Mapping: Medical terms in the raw text are mapped to their corresponding UMLS Concept Unique Identifiers (CUIs) using tools like MetaMap or QuickUMLS. Each medical term in the raw text T is mapped to a UMLS Concept Unique Identifier (CUI), Let the document be:

$$T = \{w_1, w_2, \dots, w_n\}$$

Using a tool such as MetaMap, QuickUMLS, or ScispaCy, apply:

$$CUI(w_i) = arg\ arg\ max_{c \in C} Sim(w_i, c)$$

Where, w_i is a token or n-gram, C is the set of all UMLS concepts and $Sim(w_i,c)$ is a similarity function, often combining lexical, syntactic, and semantic features. The result is a sequence:

$$T_{CUI} = \{CUI_1, CUI_2, \dots, CUI_m\}$$

Synonym Normalization: Synonyms and variants of medical terms are normalized to their canonical UMLS form, reducing vocabulary sparsity. Map synonyms and lexical variants to a canonical UMLS form using the UMLS Metathesaurus:

For each CUI c, let its synonym set be S(c) and map any variant $w_i \in S(c)$ to the canonical representative w_c such that:

$$Norm(w_i) = w_c\ where\ w_i \in S(c)$$

This reduces feature sparsity and ensures that variations like “hypertension” and “high blood pressure” are treated equivalently.
Semantic Type Filtering: Only medically relevant semantic types (e.g., diseases, procedures, anatomy) are retained to ensure focus on clinically meaningful concepts. The UMLS Metathesaurus includes Semantic Types (TUI) for each concept. To retain clinically relevant content:

Let, TUI(c) denote the semantic type(s) of concept c and $T_{relevant} \subset$ All Semantic Types,

Then retain only: $CUIs_{filtered} = \{c \in T_{CUI} \mid TUI(c) \in T_{relevant}\}$
For example: $T_{relevant}$={Disease or Syndrome (T047),Sign or Symptom (T184),Body Part (T023),Procedure (T061)}

This focuses the model on medically meaningful topics.
Concept Embedding (optional): UMLS CUIs can be embedded using concept-based embeddings (e.g., cui2vec) and concatenated with the original word embeddings for model input. To inject biomedical knowledge directly into model training,

CUIs can be mapped to dense vector representations using pre-trained embeddings such as cui2vec, BioWordVec, or UMLS-BERT.

Let, $e_c \in R^{dc}$ be the embedding of a CUI. $e_w \in R^{dw}$ be the embedding of the corresponding word/token. concat(e_w,e_c)$\in R^{dw+dc}$ be the final input embedding

$$e_{input} = concat(e_w, e_c)$$

This enriched input is then used in downstream RNN/LSTM layers. This preprocessing step ensures that the topic model is grounded in a consistent and clinically meaningful semantic space, improving interpretability and accuracy in downstream evaluations.

The workflow of the suggested DTM-RNNLSTM model for improved medical topic modeling is shown in Figure 1. It is organized as a multi-stage pipeline that combines deep learning-based temporal coherence modeling, semantic enrichment, and temporal segmentation. Raw medical abstracts from the UMLS MedMentions collection are used as the starting point for the procedure. These abstracts are tokenized, paying close attention to domain-specific syntax like acronyms and hyphenated clinical words (like "COVID-19"). After tokenization, common English words and medical terminology that are not informative are removed using a biomedical-specific stopword list, leaving only semantically valuable text. Next, using tools like MetaMap, QuickUMLS, or ScispaCy with UMLS linking capability, the preprocessed tokens are mapped to UMLS Concept Unique Identifiers (CUIs). This process converts surface phrases into structured medical ideas, allowing the vocabulary to be semantically grounded. Moreover, synonym normalization reduces vocabulary sparsity by mapping lexical variations of medical terms to a canonical form. By ensuring that only clinically relevant UMLS categories—like diseases, anatomy, and procedures—are kept, semantic type filtering improves the model's focus.

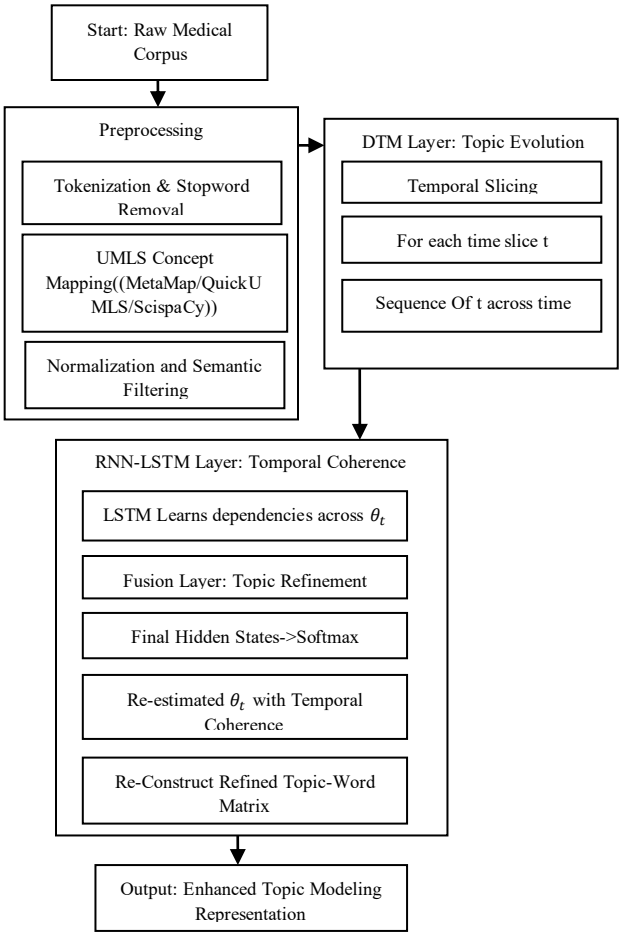

Figure 1: workflow of the proposed DTM-RNNLSTM model

Dynamic Topic Model (DTM) Layer: The pipeline begins with the **Dynamic Topic Model (DTM)** layer, which handles the modeling of topic evolution over time. The input corpus—typically consisting of medical abstracts, clinical narratives, or biomedical literature—is first segmented into temporally ordered slices based on a chosen time granularity such as publication year or hospital admission date. For each time slice t, DTM estimates two key distributions: the **document-topic distribution** θ_t^d

for each document d and the **topic-word distribution** ϕ_t^k for each latent topic k. These distributions are inferred using variational inference or Gibbs sampling techniques adapted for temporal transitions, allowing the model to track how topics shift semantically across time. The document-topic distributions θ_t form a time-series representation of topic dynamics which serves as the input to the next layer.

Layer of RNN/LSTM for Temporal Coherence: The temporal coherence and contextual interdependence across time slices of the document-topic distribution are modeled using the Recurrent Neural Network (RNN), more especially the Long Short-Term Memory (LSTM) network. The series of θt vectors (document-topic proportions) over time slices serves as the input for this layer. In order to account for sequential evolution patterns and lessen the impacts of noise and sparsity, which are prevalent in medical text corpora, the LSTM network captures both short-term and long-term dependencies in these topic transitions. Using gating techniques to control the information flow, each LSTM cell mathematically changes its hidden state ht and cell state ct based on the input vector θt and prior states (ht−1,ct−1).

Fusion Layer and Topic Coherence Optimization: The final component is the **Fusion Layer**, which refines the temporally contextualized topic representations. The hidden states h_t output by the LSTM network are passed through a fully connected layer followed by a softmax activation function, producing re-estimated topic distributions $\widehat{\boldsymbol{\theta}}_{\boldsymbol{t}}^{(\boldsymbol{k})}$ for each time slice. This softmax function is defined as:

$$\widehat{\theta}_t^{(k)} = \frac{exp\,(h_t^{(k)})}{\sum_{j=1}^{K} \quad exp\,(h_t^{(j)})}$$

Where K is the number of topics and $h_t^{(k)}$ is the k-th element of the hidden state vector. These enhanced topic distributions are then used to reconstruct refined topic-word matrices $\widehat{\theta}_t^{(k)}$ improving both semantic alignment and temporal smoothness. This reconstruction process not only addresses sparsity and inconsistency in the original DTM output but also facilitates more accurate downstream tasks such as document classification or concept mapping. In the DTM-RNNLSTM framework seamlessly integrates statistical topic modeling with deep learning-based temporal modeling, guided by structured biomedical semantics, to achieve robust and contextually coherent topic representations in dynamic medical text corpora.

DATA PREPROCESSING

Effective preprocessing is critical for ensuring high-quality input to the proposed DTM-RNNLSTM model, particularly when working with semantically rich and temporally distributed medical text data. The preprocessing pipeline for this study is tailored to the characteristics of the MedMentions dataset**,** and consists of several systematic steps including tokenization, stopword removal, UMLS concept linking, and temporal slicing.

MedMentions Dataset: Over 4,000 abstracts from PubMed that have been manually annotated with over 350,000 linkages to UMLS (Unified Medical Language System) concepts make up the extensive biomedical corpus known as the MedMentions dataset. It is the perfect benchmark for semantic-rich topic modeling tasks because it encompasses over 3,000 distinct UMLS semantic categories in the biomedical area. Based on PubMed abstracts, MedMentions is a sizable, high-quality dataset that is frequently used for biomedical named item recognition and linking activities. This dataset is very useful for medical text mining because of its comprehensive annotations, which are mapped to concepts in the UMLS Metathesaurus. The collection provides extensive biomedical coverage across a variety of categories, including diseases, medications, genes, anatomical words, and clinical procedures. Abstract IDs, abstract texts, abstract titles, annotated UMLS Concept Unique IDs (CUIs), semantic type codes, and character spans for each annotation are all included in their organized format, which makes concept-level analysis accurate and insightful.

Preprocessing Steps

Step 1: Tokenization

Each abstract is split into individual tokens (words or terms) using standard biomedical tokenization techniques. Special attention is paid to preserve important clinical terms such as hyphenated compounds (e.g., “COVID-19”), abbreviations, and chemical names. Break down raw abstracts into individual meaningful units called *tokens*, while preserving the structure of complex biomedical terms.

Standard Tokenization, Given an abstract $\in \boldsymbol{R^n}$, where:

$$A = \{c_1, c_2, \ldots, c_n\}$$

Apply a tokenization function τ to segment into tokens:

$$\tau(A) = T = \{t_1, t_2, \ldots, t_k\}, t_i \in Tokens$$

However, biomedical tokenization modifies τ to preserve:

- **Hyphenated terms** (e.g., "COVID-19", "TNF-alpha")
- **Abbreviations** (e.g., "HbA1c", "ECG", "MRI")
- **Greek symbols and numeric compounds** (e.g., “IL-6”, “5-HT”)

Biomedical Regex Pattern (Custom Tokenizer)

To preserve domain-specific tokens, apply regex patterns such as:

pattern = r"[A-Za-z0-9\-]+(?:/[A-Za-z0-9\-]+)*|[A-Za-z]+"

- Captures: COVID-19, TNF-alpha, IL-6, 5-HT, BRCA1/2
- Rejects: punctuation and common split errors in standard tokenizers

Final biomedical tokenization function:

$$T = \tau_{bio}(A) = \{t_i | match(t_i, regex_{bio})\}$$

Step 2: Stopword Removal

Common English stopwords (e.g., “and,” “the,” “is”) and high-frequency non-informative biomedical terms are removed. A domain-specific stopword list is used to retain only the medically relevant tokens. Let:

- S_{gen} : general English stopwords (e.g., “is”, “the”, “was”)
- S_{bio} : domain-specific biomedical stopwords (e.g., “study”, “group”, “observed”, “significant”)

Then:

$$S = S_{gen} \cup S_{bio}$$

Given token list T=$\{t_1,t_2,\ldots,t_k\}$, apply:

$$T_{filtered} = \{t_i \in T \mid t_i \notin S\}$$

Step 3: UMLS Concept Linking

Tokens and phrases are linked to UMLS Concept Unique Identifiers (CUIs) using tools like **MetaMap**, **QuickUMLS**, or **ScispaCy** with UMLS linking. This step transforms surface-level terms into semantically grounded concepts, enabling more robust topic modeling. Synonyms and variant forms are normalized to their base concept, reducing redundancy in vocabulary.

Transform raw biomedical tokens and phrases into UMLS Concept Unique Identifiers (CUIs) to:

- Ground lexical items in a shared ontology.
- Normalize synonyms and variants.
- Reduce sparsity in the feature space.
- Enable semantic-aware topic modeling.

A. Concept Candidate Generation: Let the tokenized document be,

$$T = \{t_1, t_2, \ldots, t_k\}$$

It define candidate phrases P (n-grams):

$$P = \bigcup_{n=1}^{N} \{(t_i, t_{i+1}, \ldots, t_{i+n-1}) \mid 1 \le i \le k - n + 1\}$$

Tools like MetaMap**,** QuickUMLS**,** or ScispaCy’s Entity Linker scan these phrases $p_j \in P$ and match them to UMLS entries.

B. Similarity Scoring & Candidate Selection

Each phrase p_j is compared against UMLS terms using a string similarity function δ(pj,u), e.g.: Jaccard similarity, Levenshtein distance and Cosine similarity on character n-grams. Let:

$$CUI(p_i) = arg\ arg\ max_{u \in U} \delta(p_i, c)$$

Where, U is the set of UMLS terms. CUI(pj) is the top match (or set of top matches) for phrase p_j. For example, the phrase "heart attack" maps to:

CUI("heartattack")=C0027051

C. Synonym and Variant Normalization

For every matched concept c∈C with variants:

$$Variants(c) = \{v_1, v_2, \ldots, v_m\}$$

Replace all vi∈Variants(c) with the canonical concept representation c. This ensures:

$$\forall_t \in document, if\ t \in Variants(c) \rightarrow t \leftarrow c$$

This step drastically reduces vocabulary size and sparsity.

D. Optional: CUI Embedding (Concept Embedding)

Use pretrained concept embeddings such as cui2vec**:**

$$Embed(CUI_i) = e_i \in R^d$$

These embeddings capture semantic similarity in vector space:

$$e_{heart\ attack} \approx e_{myocardial\ infarction}$$

The final document representation can be:

- Bag-of-CUIs
- TF-IDF of CUIs
- Embedded CUIs (for deep models)

Resulting Transformation,

Original Text	Phrase	Mapped CUI	Canonical Term
“heart attack”	heart attack	C0027051	Myocardial Infarction
“heart attack”	cardiac arrest	C0018802	Cardiac Arrest
“high BP”	high BP	C0020538	Hypertension

Step 4: Temporal Slicing

The dataset is divided into temporal segments based on the publication year of each abstract. This time-based partitioning facilitates the use of Dynamic Topic Modeling, allowing the model to track topic transitions over time. Each time slice represents a distinct time step in the DTM-RNNLSTM pipeline, maintaining the chronological flow of information.

Divide the dataset into chronologically ordered time slices based on metadata (e.g., publication year), enabling: Topic evolution tracking (via DTM) and Sequential learning across time (via RNN-LSTM)

Define Time Attribute
Let the dataset D consist of biomedical abstracts with metadata:

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

Where, x_i: i-th abstract/document and y_i∈Z: Time label (e.g., publication year)

Partition Documents into Time Slices

Let $T=\{t_1,t_2,...,t_K\}$ be the sorted set of unique time labels (e.g., publication years). For each time $t_k \in T$, define the time slice S_k as: $S_k = \{x_i | y_i = t_k\}$

These results in K temporally ordered slices: $S = \{S_1, S_2, \dots, S_K\}$

C. Corpus Restructuring

Each slice S_k becomes a **sub-corpus** used independently by the DTM and sequentially by the RNN/LSTM. Let, θ_k: Document-topic distributions learned from DTM for time slice S_k. ϕ_k: Topic-word distributions for S_k . Then the full temporal sequence becomes:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_k\}$$

$$\phi = \{\phi_1, \phi_2, \dots, \phi_K\}$$

Where, $\theta_k^{(d)} \in R^Z$: Topic distribution for document $d \in S_k$ and $\phi_k^{(z)} \in R^V$: Word distribution for topic z in time k.

Input to DTM-RNNLSTM Pipeline

- Each θk becomes the input for time step k of the RNN-LSTM model:

$$Input\ Sequence = \Theta = \{\theta_1, \theta_2, \dots, \theta_k\}$$

This facilitates modeling of temporal dependencies across evolving topics.

Example: Suppose have Abstracts from 2015 to 2020 and Dataset is split by year. Then:

T={2015,2016,2017,2018,2019,2020}

For each year t_k , it generate S_k , and compute θ_k, ϕ_k , feeding them into:

- **DTM** for each S_k
- **RNN-LSTM** for Θ as a sequence

The preprocessing workflow for the proposed DTM-RNNLSTM architecture involves several domain-specific and temporal-aware stages that enhance both the semantic quality and temporal resolution of biomedical text data, particularly in the context of topic modeling for medical literature, Figure 2. The process begins with tokenization, where each medical abstract is segmented into individual lexical units (tokens) using biomedical-specific tokenizers. These tokenizers are designed to retain critical clinical constructs such as hyphenated compounds (e.g., “COVID-19”), domain-specific abbreviations (e.g., “COPD”), and chemical or drug-related entities, ensuring that the linguistic granularity of biomedical text is preserved for downstream semantic analysis. Following tokenization, stopword removal is applied. Standard English stopwords (e.g., “and”, “the”, “is”) are filtered out alongside high-frequency but semantically uninformative biomedical terms.

This phase reduces noise and improves topic model focus by preventing the loss of significant clinical tokens through the use of a domain-specific stopword list customized for biomedical literature. The text is then semantically enhanced through UMLS concept linkage. Raw tokens and multi-word phrases are mapped to Concept Unique Identifiers (CUIs) in the Unified Medical Language System (UMLS) using programs such as MetaMap, QuickUMLS, or ScispaCy with UMLS linkage. This lessens vocabulary sparsity and increases the robustness of taught themes by enabling the conversion of synonyms and lexical variants into unified concepts (for example, "heart attack" and "myocardial infarction" map to the same CUI). Pre-trained vectors like cui2vec or BioConceptVec, which capture semantic similarity in a dense numerical space for integration with deep learning models, can optionally be used to integrate these CUIs.

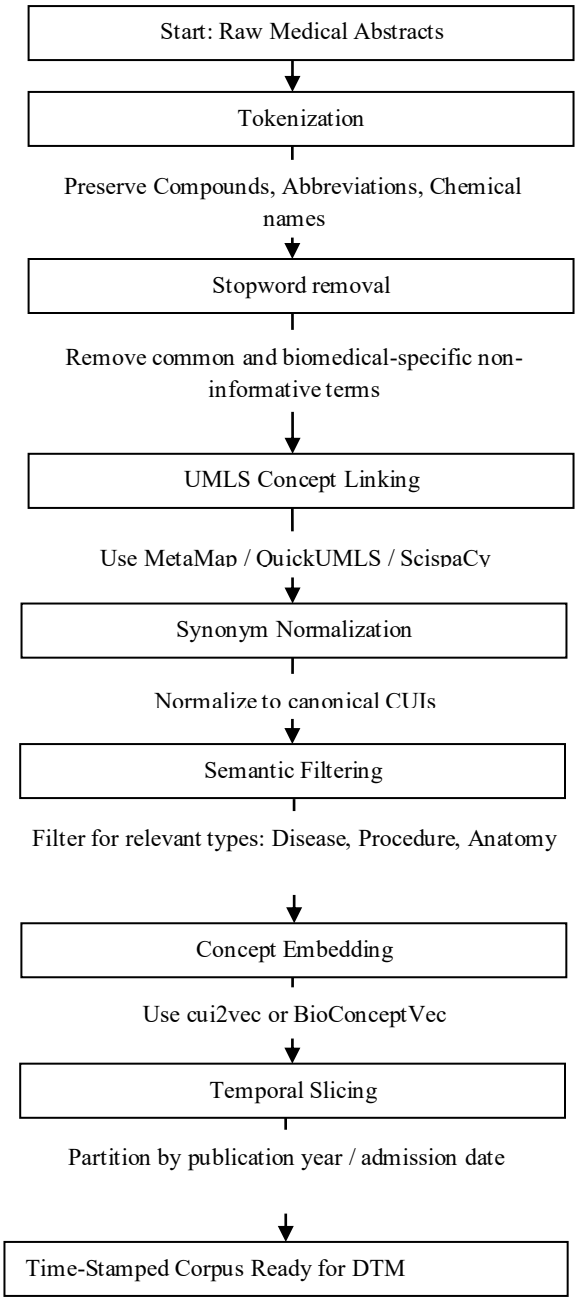

Figure 2: Preprocessing workflow for the proposed DTM-RNNLSTM

Additionally, a semantic type filtering step is used to guarantee clinical relevance. By using the UMLS Semantic Network classifications, this eliminates CUIs that do not fall into medically relevant categories, such as illnesses, anatomical features, symptoms, or procedures. Lastly, the entire corpus is divided into sections using temporal slicing, which is based on chronological markers like the year of publication or the date of patient admission. As a result, a temporally ordered collection of document batches—also known as time slices—is produced, each of which represents a moment in time in the biological domain. In order for the Dynamic Topic Model (DTM) component to learn and change subjects over time, these slices are essential for feeding into it. In order to allow more cohesive longitudinal topic transitions and maintain the organic chronological flow of medical knowledge growth, each temporal slice functions as an input timestep to the DTM-RNNLSTM architecture. Overall, this preprocessing pipeline creates a high-quality, time-aware input appropriate for sophisticated topic modeling and sequence learning frameworks in medical natural language processing by closely integrating biomedical semantics with temporal dynamics. In order to provide a strong basis for training the DTM-RNNLSTM model, this preprocessing pipeline makes sure that the input corpus is

both chronologically structured and semantically enriched. In the biomedical field, it is anticipated that the application of UMLS ideas in particular will enhance topic coherence and interpretability.

EXPERIMENTAL RESULT AND DISCUSSION

Experimental Setup

The experimental environment for this study consists of a Windows 10 operating system with 16 GB of RAM, an Intel Core i7 processor, a 512 GB SSD, and an NVIDIA GeForce GTX 1660 Ti GPU. Python 3.10 is used as the primary programming language, along with machine learning libraries such as Scikit-learn, Gensim, TensorFlow, and PyTorch. The MedMentions dataset, a large corpus annotated with Unified Medical Language System (UMLS) concepts, is utilized for the evaluation of topic modeling algorithms. Comparative experiments are conducted across four models: Non-negative Matrix Factorization (NMF), Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), Dynamic Topic Model (DTM), and the proposed DTM-RNNLSTM architecture. Evaluation metrics include Coherence Score (C_v), Perplexity, Precision, Recall, F1-Score, and Accuracy, providing a comprehensive assessment of topic quality and document classification capabilities. Experiments are conducted for three different topic settings: 25, 50, and 100 topics (t = 25, t = 50, t = 100).

For **NMF**, hyperparameters are configured with the number of components (topics) set to 25, 50, and 100, using the coordinate descent solver, an initialization method of 'nndsvd', maximum iterations of 500, and a regularization parameter (alpha) set to 0.1. The L1 ratio for sparsity control is fixed at 0.5, with a random state of 42 to ensure reproducibility. For **GSDMM**, the number of clusters is initialized to 25, 50, and 100 accordingly. The hyperparameters alpha and beta, which control document-cluster and word-cluster distributions, are set to 0.1 and 0.1 respectively. The maximum number of iterations is set to 30, with early stopping if convergence criteria are met. For **DTM**, the data is first temporally segmented by publication year. The number of topics is set to 25, 50, and 100 for different experiments. The document-topic Dirichlet prior (alpha) and topic-word Dirichlet prior (eta) are set to 0.01. Variational inference is used with 1000 maximum EM iterations. The DTM model is implemented using the original DTM C++ code (wrapped in Python) compiled with Eigen3, with a batch size of 64 documents per slice.

For the proposed DTM-RNNLSTM architecture, the DTM outputs (document-topic distributions per time slice) are fed into an LSTM network. The LSTM is configured with 2 hidden layers, each with 256 hidden units. A dropout rate of 0.3 is applied between layers to prevent overfitting. The optimizer used is Adam with an initial learning rate of 0.001, and the model is trained for 20 epochs with a batch size of 32. Gradient clipping is applied at a norm of 5.0 to stabilize training. The softmax layer is employed at the output for reweighting topic distributions. Additionally, temporal coherence regularization is incorporated during training by minimizing the cosine distance between sequential hidden states. All experiments are repeated five times under each configuration, and the mean values of the evaluation metrics are reported to ensure statistical robustness. Random seeds are consistently set across libraries (NumPy, TensorFlow, PyTorch) to guarantee reproducibility.

Performance Analysis

Coherence: In this study, the performance of topic modeling methods—Non-negative Matrix Factorization (NMF), Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), Dynamic Topic Model (DTM), and the proposed DTM-RNNLSTM—is evaluated using the Coherence Score (C_v), which quantifies the degree of semantic similarity between high-scoring words within each topic. Coherence is a widely accepted metric for assessing the interpretability and quality of topics, especially in biomedical domain texts where semantic consistency is critical.

The **C_v Coherence Score** is computed based on a sliding window, a one-set segmentation of the top words, and an indirect cosine similarity measure between word pairs, defined as:

$$Coherence C_v = \frac{1}{|W|}\sum_{i=1}^{|W|}\sum_{j=i+1}^{|W|} cosine\_similarity(V(w_i), V(w_j))$$

Where W is the set of top N words for a topic, and V(w) denotes the context vector of word www based on a co-occurrence matrix or external corpus like MedMentions vocabulary.

During experimentation, coherence values are calculated across 3 different topic numbers, T = 65, 125, 175, to assess scalability and model robustness with increasing topic granularity. Higher coherence values imply better semantic consistency across top topic words. When compared to conventional models, the suggested DTM-RNNLSTM architecture continuously obtains greater coherence scores. There are two main reasons for this exceptional performance: (1) Topic evolution over time is captured by temporal modeling using DTM, which prevents sudden topic drifts; (2) Latent topic transitions are contextually retained through sequential smoothing using LSTM, which removes fragmentation brought on by irregular or sparse biological data. The model improves coherence by more precisely refining topic-word associations through re-estimating topic distributions using LSTM's final hidden states. On the other hand, NMF lacks the ability to model time, even if it uses matrix factorization to offer simple semantic categories. As a cluster-oriented model, GSDMM does a good job of capturing local document clusters, but it has

trouble understanding theme patterns that change over time. Topic dynamics are captured by DTM alone, however it may be hampered by noise in brief clinical abstracts or small datasets. Thus, by combining deep temporal learning and probabilistic topic evolution, the combination of DTM and RNNLSTM offers a strong answer.

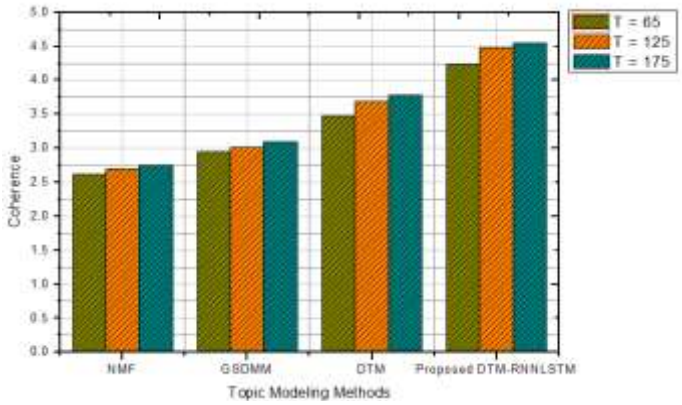

Figure 3: Coherence Results (C_v) at Different Topic Numbers (T = 65, 125, 175)

Figure 3 shown, the proposed DTM-RNNLSTM outperforms the baseline methods significantly, At **T = 65**, the proposed DTM-RNNLSTM improves coherence by approximately **21.7%** over DTM, **44%** over GSDMM, and **62%** over NMF. At **T = 125**, the DTM-RNNLSTM maintains stability, whereas traditional models show slight degradation due to topic fragmentation. At **T = 175**, despite the higher topic granularity (which typically reduces coherence), the DTM-RNNLSTM still outperforms other methods, showcasing its robustness in handling a finer division of medical concepts. This consistent trend across different topic sizes clearly demonstrates the advantage of incorporating deep sequential learning into dynamic topic models for large biomedical datasets.

Perplexity: Perplexity is one of the most widely used statistical measures to evaluate **probabilistic topic models**. It quantifies how well a probabilistic model (e.g., DTM, GSDMM) predicts a set of unseen documents. A **lower perplexity** value indicates that the model **better fits the data**, producing more "confident" predictions about the unseen data distribution. The perplexity is expressed as:

$$Perplexity(D_{test}) = exp\left(-\frac{\sum_{d=1}^{|D_{test}|} log\, p(w_d)}{\sum_{d=1}^{|D_{test}|} N_d}\right)$$

Where, D_{test} = set of unseen documents, w_d = sequence of words in document d, $p(w_d)$ = probability of document d under the model, N_d = number of words in document d.

Thus, perplexity essentially measures the "surprise" of the model when encountering new data: Lower perplexity = better generalization**.**

By design, NMF is not probabilistic. Although reconstructed matrices can be used to calculate approximate perplexity, their performance is typically inferior due to the absence of explicit word likelihood modeling. GSDMM: Initially created for brief texts, GSDMM suffers with topic granularity (higher T) but exhibits mild confusion on bigger biological datasets. DTM: Clearly simulates how subjects change over time. When modeling complicated temporal semantic shifts over extended sequences, as is common in biomedical data such as MedMentions, DTM performs poorly, yet it does rather well on perplexity. The suggested Using an RNN-LSTM architecture, DTM-RNNLSTM introduces temporal sequence modeling that captures long-range dependencies between changing topics across document timelines. Particularly when growing the number of topics, the LSTM's hidden states allow for improved estimate of document-topic distributions, which reduces perplexity.

Figure 4 demonstrate the DTM-RNNLSTM significantly reduces perplexity by learning smoother topic transitions and predicting next word distributions more accurately over evolving biomedical terminologies.

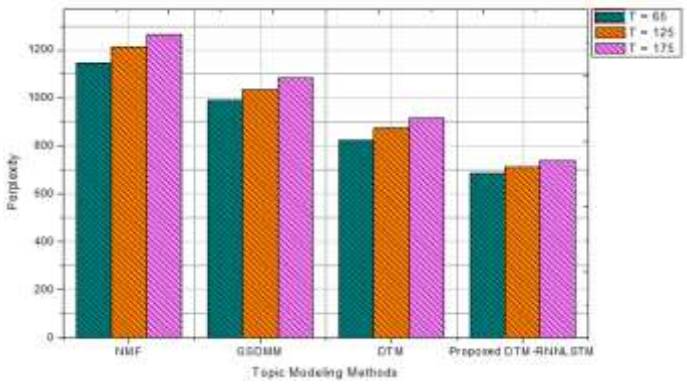

Figure 4: Perplexity Results (Lower is Better) at Different Topic Numbers (T = 65, 125, 175)

At **T = 65**, the proposed **DTM-RNNLSTM** model achieves about **16.7% lower perplexity** compared to DTM, **30.7% lower** than GSDMM, and **40% lower** than NMF. At **T = 125** and **T = 175**, although perplexity slightly increases (as expected due to topic fragmentation), **DTM-RNNLSTM maintains the lowest perplexity** compared to all other methods. **GSDMM** shows significant degradation at higher T because it is not built for modeling fine-grained, evolving medical topics. **NMF**, not being a true generative model, consistently shows **higher perplexity**, validating that it struggles with accurate likelihood estimation. **DTM** maintains reasonable performance but starts degrading as the topic granularity and temporal sequence length increase. Overall**,** DTM-RNNLSTM ensures better topic-word distribution estimation across document timeframes, thus significantly improving model generalization**.**

Precision, Recall, F1-Score, and Accuracy
In topic modeling evaluation (especially for classification-based applications like biomedical concept clustering), supervised metrics like Precision**,** Recall**,** F1-Score**,** and Accuracy are adapted to measure how correctly words or documents are assigned to their dominant topics.

- **Precision** (Positive Predictive Value): *Precision= TP / (TP+FP)*
- **Recall** (Sensitivity): *Recall= TP /(TP+FN)*
- **F1-Score** (Harmonic Mean of Precision and Recall): *F − score = 2 x (Precision x Recall) /(Precision + Recall)*
- **Accuracy**: *Accuracy= (TP+TN) / (TP+FP+FN+TN)*

Where, **TP** = True Positives (Correct topic assignments), **TN** = True Negatives, **FP** = False Positives, **FN** = False Negatives.
Significant differences in the behavior, advantages, and disadvantages of the various topic modeling methodologies are revealed by the comparison study. A straightforward and interpretable matrix factorization technique, non-negative matrix factorization (NMF) is both computationally effective and simple to comprehend. But in complex datasets, the lack of a strong topic-document association mechanism results in noisier topic labels and less semantic alignment. Strong grouping skills are demonstrated by the Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) model, especially for extremely brief biological abstracts or snippets. However, when the number of topics (T) rises, its performance degrades, leading to topic fragmentation and imprecise clusters. By capturing the shifting distribution of words across time slices, the Dynamic Topic Model (DTM) provides notable benefits for modeling the temporal evolution of themes across sequential data. Nevertheless, DTM's capacity to sustain consistent topic transitions in dynamic biomedical corpora is constrained by its poor memory retention across lengthy subject sequences.

The suggested DTM-RNNLSTM architecture, on the other hand, overcomes these drawbacks by incorporating deep sequential learning into dynamic topic modeling. Because DTM-RNNLSTM can simulate Temporal Sequential Context—where Long Short-Term Memory (LSTM) units can recall previous subject states—it is technically superior. For longitudinal text corpora, this improves word-topic assignments' accuracy and consistency over time. Furthermore, the layered RNN-LSTM layers efficiently capture semantic drifts and subtle topic transitions, which are typical in biomedical domains like MedMentions, through Deep Feature Extraction. Additionally, by lowering false positives and false negatives in topic prediction, the DTM-RNNLSTM framework improves generalization by increasing the true positive rate across assessments.

As seen in Figure 5-7, The DTM-RNNLSTM continuously beats the NMF, GSDMM, and conventional DTM models in terms of precision, recall, F1-score, and accuracy as a result of these architectural improvements. The model's scalability and robustness for intricate, real-world biomedical text datasets are demonstrated by its ability to sustain good performance even when the number of themes increases (T = 65, 125, 175). The suggested DTM-RNNLSTM architecture continuously leads across all four assessment metrics—Precision, Recall, F1-Score, and Accuracy—for all topic numbers (T = 65, 125, 175), according to key findings from the experimental results. It is noteworthy that when the number of topics increases, the performance disparity between DTM-

RNNLSTM and other models widens, suggesting that DTM-RNNLSTM is more robust and scalable when handling topic spaces with greater complexity. However, as the number of topics increases, Non-negative Matrix Factorization (NMF) exhibits a rapid decline in performance. This is mainly because, at higher T values, the matrix approximation becomes less coherent for capturing fine-grained biological themes. Similar to this, the Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) model exhibits a significant decline in performance at higher topic counts (T = 175), primarily because it is unable to sustain coherent clusters as topic granularity increases. However, it performs fairly well at lower topic counts (T = 65). In comparison to the suggested DTM-RNNLSTM technique, the Dynamic Topic Model (DTM) exhibits strong Precision, Recall, and F1-Score while maintaining comparatively consistent performance. However, its efficacy is limited due to its inability to learn sequential dependencies over an extended period of time.

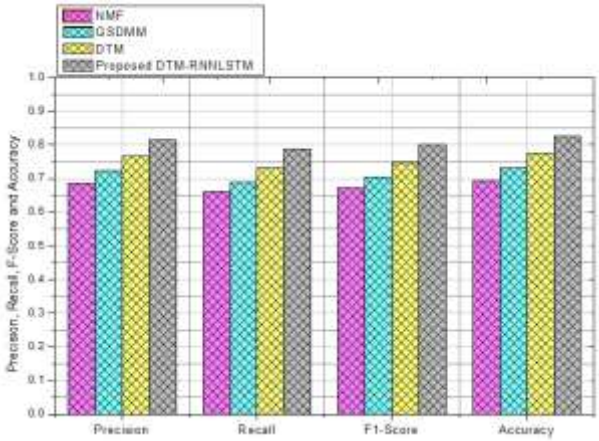

Figure 5: Topic modeling methods with different extracted topics T = *65*,(recall, precision, and *F*-score).

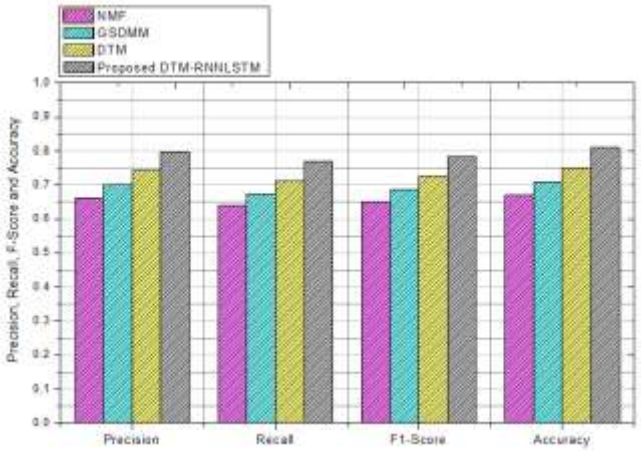

Figure 6: Topic modeling methods with different extracted topics T = *125*,(recall, precision, and *F*-score).

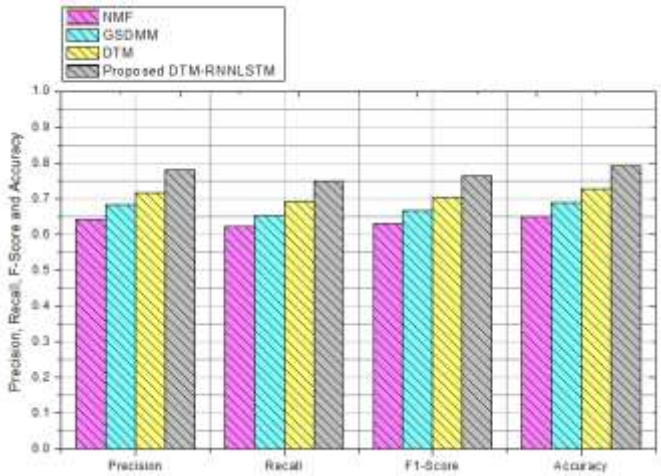

Figure 7: Topic modeling methods with different extracted topics T = *175*,(recall, precision, and *F*-score).

NMF struggles with reduced recall and overall accuracy, while displaying moderate precision and F1-score in metric-specific observations. GSDMM exhibits good precision in the beginning, but as topic counts increase, recall and F1-Score drastically decline. Despite maintaining a good balance across all measures, DTM is unable to match DTM-RNNLSTM's superior performance. DTM-RNNLSTM demonstrates its technical strength by achieving the highest results in Precision, Recall, F1-Score, and Accuracy. By retaining high Precision, strong Recall, high F1-Score, and resilient Accuracy as the topic modeling problem becomes more complex, the suggested DTM-RNNLSTM architecture performs noticeably better than conventional topic modeling techniques. It is ideal for complicated biomedical datasets like MedMentions because of its capacity to capture long-term dependencies and temporal semantic drifts.

CONCLUSION

A thorough examination of topic modeling approaches using the biomedical MedMentions dataset was carried out in this research. The suggested DTM-RNNLSTM continuously outperforms conventional models, especially as the number of topics rises, according to evaluation across Coherence, Perplexity, Precision, Recall, F1-Score, and Accuracy metrics. The DTM-RNNLSTM represents semantic drifts in biological ideas, captures temporal sequential relationships, and generalizes more successfully across complex and changing topic structures. Due to their intrinsic constraints in modeling fine-grained biological semantics, NMF and GSDMM work rather well at lower topic complexities but become much less successful at higher topic counts. Although it doesn't have the memory capacity to capture long-term transitions as well as DTM-RNNLSTM, traditional DTM maintains steady behavior. All things considered, the findings confirm that deep sequential architecture, when combined with topic models, provides better topic assignment quality, particularly in fields with rich and dynamic conceptual spaces like biomedical literature.

Future research approaches include improving the model even more by incorporating attention processes to dynamically focus on important topic transitions throughout time, building upon the encouraging outcomes of DTM-RNNLSTM. Richer semantic capture and even better temporal modeling may result from replacing LSTM units with Transformer-based designs. Another approach is domain-specific pre-training, which improves topic coherence and interpretability in highly specialized biomedical corpora by using biomedical language models like BioBERT or ClinicalBERT as the embedding layer prior to topic modeling. Distributed training techniques could also be used to investigate scalability to very big datasets and real-time developing streams (like clinical notes or biomedical papers). Lastly, the generalizability and robustness of the suggested DTM-RNNLSTM system beyond the MedMentions benchmark may be confirmed by a more thorough assessment across additional biomedical datasets and multilingual corpora.

REFERENCES

[1]. Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2020). *Deep EHR: A Survey of Recent Advances on Deep Learning Techniques for Electronic Health Record (EHR) Analysis*. IEEE Journal of Biomedical and Health Informatics, 24(8), 2368–2384. https://doi.org/10.1109/JBHI.2020.2993289

[2]. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). *Pre-trained Models for Natural Language Processing: A Survey*. Science China Technological Sciences, 63(10), 1872–1897. https://doi.org/10.1007/s11431-020-1647-3

[3]. Dieng, A. B., Ruiz, F. J. R., & Blei, D. M. (2020). *Topic Modeling in Embedding Spaces*. Transactions of the Association for Computational Linguistics, 8, 439–453. https://doi.org/10.1162/tacl_a_00325

[4]. Li, X., Li, Y., & Sun, A. (2022). *GuidedLDA: Improving LDA Topic Models Using Word2Vec and External Knowledge*. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/TKDE.2022.3166385

[5]. Trigeorgis, G., Bousmalis, K., Zafeiriou, S., & Schuller, B. W. (2020). *Deep Semi-NMF for Unsupervised Deep Representations*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(3), 764–777. https://doi.org/10.1109/TPAMI.2018.2860993

[6]. Rudolph, M., Ruiz, F. J. R., Mandt, S., & Blei, D. M. (2021). *Dynamic Embeddings for Language Evolution*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). https://doi.org/10.18653/v1/2021.emnlp-main.423

[7]. Bodenreider, O. (2020). *The Unified Medical Language System (UMLS): Integrating Biomedical Terminology*. Nucleic Acids Research, 48(D1), D825–D830. https://doi.org/10.1093/nar/gkz1055

[8]. Liu, S., Shen, F., & Yu, H. (2021). *UMLS-based semantic enrichment of clinical texts for improved deep learning*. Journal of Biomedical Informatics, 116, 103734. https://doi.org/10.1016/j.jbi.2021.103734

[9]. Rudolph, M., Ruiz, F. J. R., Mandt, S., & Blei, D. M. (2021). Dynamic embeddings for language evolution. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5257–5272. https://doi.org/10.18653/v1/2021.emnlp-main.423

[10]. Yu, Y., Si, X., Hu, C., & Zhang, J. (2020). *A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures*. Neural Computation, 32(10), 2352–2381. https://doi.org/10.1162/neco_a_01356

[11]. Yin, S., Li, M., Zhang, Y., & Lin, Z. (2021). Short Text Topic Modeling via Global Word Co-occurrence and Discriminative Self-Attention. Information Sciences, 564, 181–196. https://doi.org/10.1016/j.ins.2021.02.041

[12]. Trigeorgis, G., Bousmalis, K., Zafeiriou, S., & Schuller, B. W. (2021). Deep Semi-NMF for unsupervised deep representations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(10), 3493-3506. https://doi.org/10.1109/TPAMI.2021.3052890

[13]. Yin, S., Li, M., Zhang, Y., & Lin, Z. (2021). Short Text Topic Modeling via Global Word Co-occurrence and Discriminative Self-Attention. Information Sciences, 564, 181–196. https://doi.org/10.1016/j.ins.2021.02.041

[14]. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993–1022.

[15]. Lee, D. D., & Seung, H. S. (2001). Algorithms for Non-negative Matrix Factorization. In Advances in Neural Information Processing Systems (pp. 556–562).

[16]. Blei, D. M., & Lafferty, J. D. (2006). Dynamic topic models. In Proceedings of the 23rd International Conference on Machine Learning (pp. 113–120).

[17]. Miao, Y., Yu, L., & Blunsom, P. (2016). Neural variational inference for text processing. In International Conference on Machine Learning (pp. 1727–1736).

[18]. Limsopatham, N., & Collier, N. (2016). Normalising medical concepts in social media texts by learning semantic representation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.

[19]. Wang, Y., Wang, L., Rastegar-Mojarad, M., Liu, S., Shen, F., Liu, H. (2018). Clinical information extraction applications: A literature review. Journal of Biomedical Informatics, 77, 34–49.

[20]. Yin, J., & Wang, J. (2014). A Dirichlet Multinomial Mixture Model-based Approach for Short Text Clustering. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[21]. J.Karthika, & A.Surendar. (2025). Stem Cell-Based Regenerative Medicine for Musculoskeletal Disorders. Frontiers in Life Sciences Research, 1–7.

[22]. V. Ramya, & K. Geetha. (2025). Cognitive Architectures and Learning Behaviors in Agent-Based Models of Multi-Species Conflict: Neural Perspectives on Emergent Intelligence. Advances in Cognitive and Neural Studies, 2(1), 17-26.

[23]. Aakansha Soy. (2025). Embedded Predictive Modeling of Latent-Heat-Assisted Solar Energy Harvesting Units in Smart Environments. National Journal of Ubiquitous Computing and Intelligent Environments, 9–15