Combating Hallucination and Misinformation: Factual Information Generation with Tokenized Generative Transformer

Sourav Das, Sanjay Chatterji, and **Imon Mukherjee**
Department of Computer Science and Engineering
Indian Institution of Information Technology Kalyani
Kalyani, West Bengal, India
{sourav_phd21, sanjayc, imon}@iiitkalyani.ac.in

Abstract

Large language models have gained a meteoric rise recently. With the prominence of LLMs, hallucination and misinformation generation have become a severity too. To combat this issue, we propose a contextual topic modeling approach called Co-LDA for generative transformer. It is based on Latent Dirichlet Allocation and is designed for accurate sentence-level information generation. This method extracts cohesive topics from COVID-19 research literature, grouping them into relevant categories. These contextually rich topic words serve as masked tokens in our proposed Tokenized Generative Transformer, a modified Generative Pre-Trained Transformer for generating accurate information in any designated topics. Our approach addresses micro hallucination and incorrect information issues in experimentation with the LLMs. We also introduce a Perplexity-Similarity Score system to measure semantic similarity between generated and original documents, offering accuracy and authenticity for generated texts. Evaluation of benchmark datasets, including question answering, language understanding, and language similarity demonstrates the effectiveness of our text generation method, surpassing some state-of-the-art transformer models.

1 Introduction

Large language models have become paradigm-shifting research in natural language processing, with their outstanding abilities already demonstrated in a multitude of tasks (Zhao et al., 2023). While the concept of LLMs is not entirely new (Brants et al., 2007), they have obtained computational and performative success in the last couple of years due to the enormous growth of required hardware resources. Despite the research success, an issue with the existing LLMs is that, while continuing a conversation with any LLM-based conversational agent, the rapid shifting of topics and contexts as input prompts often lead the conversation into a logical void. In such a scenario, often some LLM-based agents (for instance, *ChatGPT*[1]) cannot grasp changing context and changing the topic to persuade for delivering an expected response. Others (for instance, *Google Bard*[2]) show multiple drafts of the response, allowing the users to choose the better response as per their liking. One or more such responses can even produce *misinformation*. Another problem being discussed quite a lot about LLMs is the *hallucination* problem (Ji et al., 2023). Here, the received answer is not structured with desirable logic and reasoning, and the flow of the specific content generation can heavily deviate. In addition, some level of intellectual knowledge of semantics and queries is required to explicitly fine-tune the conversation initiation questions in certain ways to get the desired answer. This knowledge is known as prompt engineering.

In this paper, we develop an incremental-learning-based contextual topic modeling algorithm, for the generation of contingently relevant information from a large corpus of research papers related to COVID-19. The use case of our approach is made in the ground truth of the scientific literature on COVID-19, where effective information generation and analysis are vital for understanding and communicating critical insights. We propose *Contextual LDA* (Co-LDA), to extract contextual topics from the training set. We generate four sets of contextually rich topics (*T1* to *T4*) with ten topic words each using the Context Scores derived from the Co-LDA algorithm, categorized into distinctive categories such as *Medical*, *Social*, *Research*, and *Generic* topics for distinct grouping.

Upon selecting the best iteration with the highest context score, we track the original sentences from the training set containing topic words from the resultant topic set. Then a benchmark dataset is ac-

[1]https://openai.com/blog/chatgpt
[2]https://bard.google.com/

Proceedings of the Joint 3rd NLP4DH and 8th IWCLUL, pages 143–152

quired as a test set and the same steps are performed as with the training set. To generate informative sentences, we construct a tokenizer-based transformer on the existing GPT-3 and call this model the *Tokenized Generative Transformer* (TGT). The extracted topic words are then converted to corresponding masks using the Context Scores before feeding them as inputs in the TGT model. The masked topic words are then converted into a numerical encoded sequence of tokens suitable for model input. This process enables our model to understand and generate contextual fact-based text with better token-level accuracy and semantics.

We evaluate the performance of generated sentences based on the accuracy in terms of contextuality and semantics. Here, we propose the *Perplexity-Similarity Scores* to calculate the pairwise similarity scores between the comparing document sets of original sentences and generated sentences. To improve the accuracy of our computations, variable-length functions are considered for the comparable sentences. A higher matching score indicates better semantic similarity, with more accurate information augmentation. Our experimental framework outperforms most of the compared baseline state-of-the-art LLMs in factual information generation tasks on the same test data.

In this work, we make several fundamental contributions, outlined as:

- We propose **Contextual LDA (Co-LDA)**, a topic modeling algorithm based on incremental learning from data and addressing limitations of the traditional LDA by emphasizing context for more meaningful topic representations.
- We introduce **Tokenizer-based Generative Transformer (TGT)** for information sentence generation, leveraging GPT-3 to overcome hallucination without extensive prompt engineering.
- We also introduce **Perplexity-Similarity Scores** for evaluating accuracy and similarity between original and generated information using variable pairwise distance computation.
- The benchmarking procedure includes evaluation and analysis of our proposed system, based on multiple standard metrics on the public benchmark corpus for the performance demonstration with the comparable SoTA language models.

2 Motivation

Exhaustive research on LLMS has exposed challenges in mitigating hallucination (Ye et al., 2023) and misinformation generation (Pan et al., 2023). Existing LLMs exhibit difficulties in maintaining logical coherence during contextual shifts, leading to misinformation (Karpinska and Iyyer, 2023).

Our motivation lies in addressing these challenges by proposing a method tailored for accurate information generation, particularly in the context of COVID-19 research literature at present. We recognize the limitations of LLMs, notably the hallucination problem (Ji et al., 2023), prompting the development of a solution that overcomes these issues.

To achieve this, we introduce Co-LDA, a contextual topic modeling algorithm. Co-LDA enhances the accuracy of topic representation by considering context, thereby improving the relevance of generated information. By leveraging Co-LDA in conjunction with our Tokenized Generative Transformer (TGT), based on GPT-3, we alleviate the need for extensive prompt engineering, addressing the challenges posed by contextually shifting topics.

The chosen methodology ensures that our model comprehensively captures the nuances of context, leading to more coherent and accurate information generation. Through the integration of Co-LDA and TGT, we aim to provide a robust solution to the challenges associated with hallucination and misinformation in the ground truth of the COVID-19-based information generation.

3 Related Works

The recent flow of NLP research has produced high-standard works for multitask applications. In this section, we follow three major paradigms that closely line up with our work.

3.1 Topic Modeling

The methodologies for topic modeling are similar to dimensionality reduction techniques used for mathematical information. It is often seen as a way of extracting the desired part from the vocabulary. The method of recognizing, modeling, and extracting the topics from any corpus can also result in the thematic representation of the data.

Latent Dirichlet Allocation (2001) is one of the most profound methods for statistically extracting topics from texts (Blei, 2001). It is a measurable

graphical model used to establish connections between different reports in the corpus. It is built using *Variational Exception Maximization* (VEM) computations to obtain the most extreme probability measures from the entire text corpus. This can usually be resolved by carefully choosing the most important words. This model follows the idea that each dataset can be represented by a probabilistic propagation of topics and each point can be represented by a probabilistic propagation of words.

Short text topic modeling is the identification of underlying topics within a collection of short text documents. One of the biggest challenges when modeling short text topics is data sparsity. Fewer words appear in the data at any one time, making it difficult to learn the relationships between words and topics. Some researchers introduced the *Topic-Semantic Contrastive Topic Model* (TSCTM), a new framework for modeling short text topics (Wu et al., 2022). TSCTM uses a contrastive learning method to learn relationships between words and topics. This contrasting learning method refines word and topic representations, strengthens the learning signal, and alleviates data sparsity issues.

3.2 Language Generation

The *Generative Pre-trained Transformers* (GPT) (2018) (Radford et al., 2018) has made a breakthrough in NLP. GPT models[3] have achieved remarkable success in various NLP tasks. GPT is from the family of the transformer networks (Vaswani et al., 2017), and comes under the group of Large language models (Zhao et al., 2023). The concept of pre-training a transformer on a large text corpus and fine-tuning the same for a specific task led to the development of these models. The core idea behind the GPT model is to use unsupervised learning to provide a foundation for language understanding, which can then be adapted to various streamlined tasks. Current research potentials in GPT involve addressing biases, improving fine-tuning techniques, and adjusting goals before training.

Large language models are effective for a variety of tasks, recognizing their tendency to generate false information. some researchers focused on assessing LLM's preference for fact-consistent content and introducing a *Factual Inconsistency Benchmark* (FIB) in the context of summarization (Tam et al., 2023). FIB compares LLM results to compare summaries of news articles that are factually consistent with summaries of news articles that are inconsistent. They used human-generated reference summaries that are reviewed for factual consistency and annotated summaries produced by summarization models that are known to be factually inconsistent. The model's accuracy in assessing factual consistency is measured by the proportion of documents that are assigned a high score for a consistent summary of facts.

3.3 Semantic Similarity

Reimers and Gurevych (2019) presented the *Sentence-BERT*, a method for generating sentence embeddings using Siamese BERT networks (Reimers and Gurevych, 2019). The authors used a pre-trained BERT model to encode sentences and learn sentence representations. Using a Siamese network architecture, the model captured semantic similarities between pairs of sentences. Their work contributed to the advancement of sentence embedding techniques for similarity learning. Using a Siamese network architecture, the model with pre-trained BERT learned how to assign sentence pairs to a similarity space in which similar sentences are grouped.

In NLP tasks such as semantic similarity assessment, paired texts often overlap and share components, making accurate semantic assessment difficult. Traditional semantic metrics based on word representations can be confounded by such overlap. To alleviate this problem, Peng et al. (2023) introduced a mask and prediction approach (Peng et al., 2023). Identify words in the *Longest Common Sequence* (LCS) as neighborhood words and predict their position distribution using *Masked Language Modeling* (MLM) from a pre-trained language model. *Neighboring Distribution Divergence* (NDD) metrics quantify semantic distance by measuring the divergence between distributions within overlapping segments.

4 Methodology

We first construct a corpus from a large collection of COVID-19 research literature and perform incremental learning to modify learning features after each topic modeling cycle. We then propose Co-LDA, a contextual topic modeling technique, to extract semantically meaningful topics catego-

[3]https://platform.openai.com/docs/guides/gpt

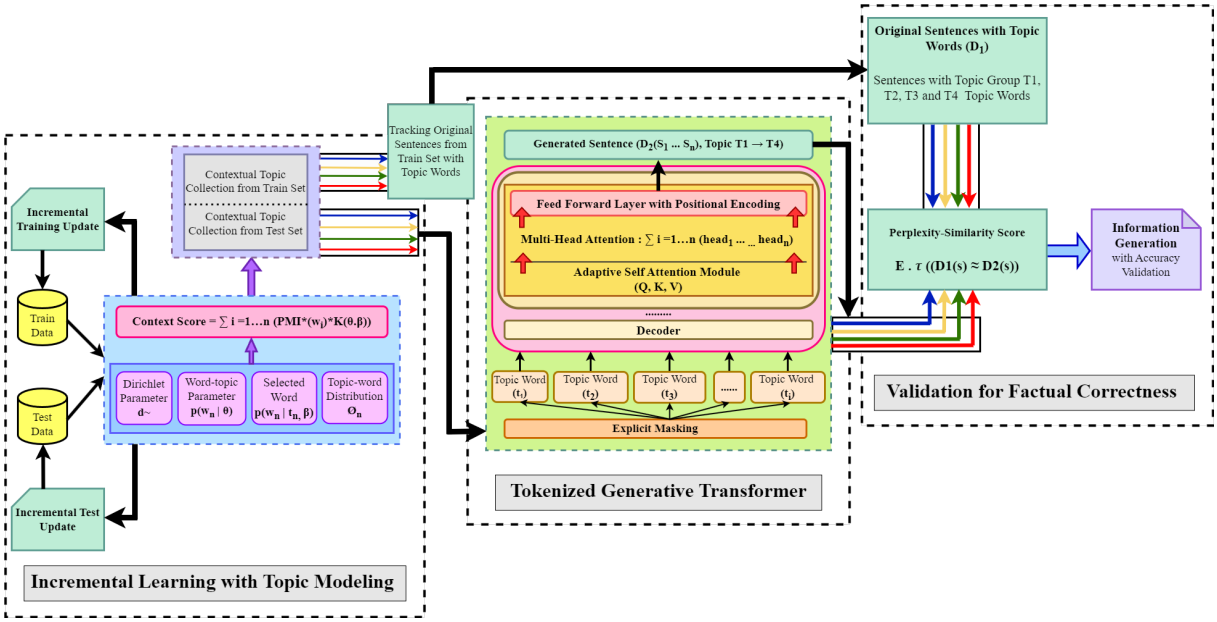

Figure 1: Overview of the proposed system framework. Each phase of the experiment is represented by separating dotted lines. Phases are segregated in order of topic modeling, information generation, and information validation. In phases 1 and 3, the respective topic streams are denoted in identifying shades as Topic T1, Topic T2, Topic T3, and Topic T4.

rized into four groups. These contextual topics are utilized to mask the Tokenized Generative Transformer (TGT), built over GPT-3, to generate factually consistent sentences. To evaluate, we introduce Perplexity-Similarity Scores to measure semantic similarity between the original and generated sentences. Multiple benchmarks demonstrate the effectiveness of our approach over comparable SoTA models.

In this section, we explain the end-to-end experiment pipeline for our work. The overview of the schematic architecture is shown in Figure 1.

4.1 Training Dataset

To construct the training dataset, we gather 5000 research papers denoted as $\mathcal{P} = \{P_1, P_2, \ldots, P_{5000}\}$ from the *arXiv* repository[4], focusing on COVID-19 research published between March 2020 to July 2023. We employ a publication retrieval framework that leverages the arXiv API to conduct a targeted search based on the user-specified topic, denoted as Topic. The topic is interactively provided through the input prompt. The search process is governed by a set of parameters represented as $\mathcal{S} = \{\text{Topic}, \text{MaxResults}, \text{SortCriterion}, \text{SortOrder}\}$, where:

- Topic is the user-specified topic (e.g., COVID-19).
- MaxResults is the maximum number of results to be retrieved (500).
- SortCriterion is the sorting criterion for the search results (e.g., submission date).
- SortOrder is the order in which the results are sorted (e.g., descending with the availability year on arXiv).

The search query returns a set of results denoted as $\mathcal{R} = \{R_1, R_2, \ldots, R_{500}\}$, where each R_i contains metadata and information about a paper, including:

- Title(T_i): The title of the paper P_i.
- Date(D_i): The publication date of paper P_i.
- ID(ID_i): The unique entry ID assigned to paper P_i by the arXiv repository.
- Summary(S_i): A concise summary of the content of the paper P_i, providing insights into its research focus.
- URL(U_i): The URL linking to the original version of the paper P_i, facilitating access to the full text for further examination.

We process these results and organize them in a structured format for subsequent analysis and exploration.

To manage the data, we initialize an empty list $\mathcal{L}$ to serve as temporary containers for each paper. We then iterate through the search results $\mathcal{R}$ and for each paper R_i, the temporary container $Container(R_i)$ is populated with the respective metadata.

[4]https://arxiv.org/

Subsequently, each Container(R_i) is appended to a new, initially empty list denoted as $\mathcal{L}$. Once all the search results have been processed, a data frame is created with the collected data, using column names corresponding to the extracted attributes. The resulting data frame is denoted as *DataFrame*($\mathcal{L}$).

As new information is continuously surfacing in terms of experiments and results from scientific literature, we utilize additional incremental units for the dataset for continuous learning. When new information is fetched during the API call, the model can be adjusted to learn from the extracted topics without having to completely retrain it from scratch. we assume to represent DataFrame($\mathcal{L}$) as a collection of data points:

$$\text{DataFrame}(\mathcal{L}) = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \quad (1)$$

Here, x_i represents the train (and test) data and y_i represents the previously discussed parameter labels associated with the data. Our goal in incremental learning is to update the parameters Φ using a new subset of data frame $\mathcal{L}_{new}$ while retaining the knowledge learned from previous API calls.

Our key challenge here is to adapt the model's parameters to the new data without forgetting the knowledge gained from the old data. We aim to minimize a loss function *J* that measures the difference between the model's predictions and the baseline parameters. In an incremental setting, we set two components for the loss:

Loss determined on the old data:

$$J_{\text{old}}(\Phi) = \sum_{i=1}^{N_{\text{old}}} \ell(f(x_i; \Phi), y_i) \quad (2)$$

Loss determined on the new data:

$$J_{\text{new}}(\Phi) = \sum_{i=1}^{N_{\text{new}}} \ell(f(x_i; \Phi), y_i) \quad (3)$$

Here, N_{old} and N_{new} represent the number of data points in the old and new datasets, respectively. The function $f(x_i; \Phi)$ represents the model's prediction for input x_i using parameters Φ, and ℓ is the loss function that quantifies the error between the model's prediction and the ground truth.

The overall objective in *incremental learning* is to find a set of updated parameters Φ^* that minimize the combined loss:

$$\Phi^* = \arg\min_{\Phi} \left(\alpha J_{\text{old}}(\Phi) + (1 - \alpha) J_{\text{new}}(\Phi)\right) \quad (4)$$

Here, α is a hyperparameter that controls the balance between preserving knowledge from the old contextual topics (J_{old}) and adapting to the new contextual topics (J_{new}).

4.2 Topic Modeling with Contextual LDA (Co-LDA)

For the improved topic modeling with context, we propose the Latent Dirichlet allocation embedded with *Context Scores* for emphasizing contextuality in extracting meaningful topics from the developed corpus. We call this scheme the Contextual LDA, or Co-LDA. Four Topic Domains or Groups[5]. are observed and derived from this method, corresponding to *T1: Medical Topic*, *T2: Social Topic*, *T3: Research Topic*, and *T4: Generic Topic*. The labeling helps us to compute context scores for different domains. We create an iterative approach to train the Co-LDA model and evaluate its context for retrieving better topic words.

Let us say the datasets containing the research paper summaries are *D*. The dataset is preprocessed to extract the text data by removing stop words, special characters, equations, and diagrams. The preprocessed dataset is denoted by *K* and each of the preprocessed summaries is denoted by k_i. We limit the number of words to k_i a maximum of 500.

$$K_{ij} = \textit{count}(j \textit{ in } k_i) \quad (5)$$

where K_{ij} represents the matrix containing $i \times j$ vector representations of the topic words from the training set.

The context score for the topic words is a measure of how well the words from each topic group or domain are related to each other, as well as to the theme of the summary. The context score is typically calculated using a measure of semantic similarity between words. We utilize the *Pointwise Mutual Information* (PMI) (Bouma, 2009) metric to perform this. Higher PMI scores indicate more contextual and meaningful topics. The context score for a set of topics is computed from the PMI score as follows:

$$\textit{Context Score} = \sum_{i=1\ldots n}^{N} (PMI(w_i) * K(\theta.\beta)) \quad (6)$$

PMI(w_i) represents the PMI score for the topic word w_i. Furthermore, θ is the probability of defining a topic belonging to a summary, and β is the

[5]We may interchangeably use the terms *topic domains* and *topic groups* throughout the paper.

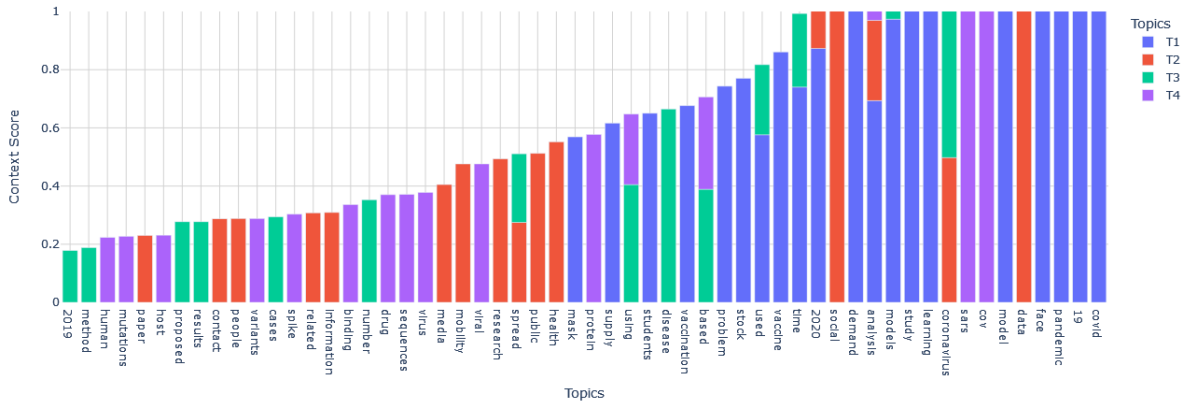

Figure 2: Topic words shown in linear incremental order in terms of Context Scores. The overlapped colors in particular bars represent corresponding topic words that are present in multiple topic groups. It ensures the context sensitivity of the proposed contextual topic modeling.

topic word distribution value. Figure 2 shows a linear increment of PMI scores for a set of topic words from all groups. We record the top ten topic words from each topic group according to the Context Scores for further analysis. The four topics with corresponding topic words are shown in Table 1.

Topic Groups	**Topic Words**
Topic 1: Medical	pandemic, epidemic, vaccine, GSA, virus, health, disease, infected, booster, death
Topic 2: Social	social, distance, isolation, lockdown, migration, remote, online, curfew, mask, sanitizer
Topic 3: Research	dataset, measures, count, model, analysis, prediction, simulation, optimization, spread, results
Topic 4: Generic	approach, crisis, initiatives, education, precaution, spread, resilience, transport, efficiency, paper

Table 1: Group-wise top ten topic words.

4.3 Sentence Generation

For generating sentences (information) on a similar set of topics, we select the *COVID-19 Open Research Dataset* (CORD-19) (Wang et al., 2020) public corpus as our test set. CORD-19 comprises a comprehensive collection of scholarly articles related to COVID-19. With over 200,000 articles encompassing various disciplines, it serves as a gold standard corpus for NLP and biomedical research. The dataset contains 84 million words and 16 million tokens and facilitates diverse analyses, from topic modeling to information extraction. CORD-19's metadata covers authors, affiliations, and publication dates for each included paper as well.

In our experiment, the topic words from four topic groups extracted from the training set are the input for the primary masking phase of the transformer. These topic words are masked prior to the input by encoding the Context Scores with the words for the awareness of the model. Using the decoder layer, these explicitly masked tokens serve as the base prompts for our Tokenized Generative Transformer model.

The masking technique is important for processing sequences of different character lengths from the topic words while preserving the advantages of parallel processing. The self-awareness mechanism uses attention scores from queries and keys to compute weighted sums of values to capture sequence relationships.

We utilize the GPT-3 language model as the platform for developing the Tokenized Generative Transformer model. After masking the topic words, the model imposes the self-attention mechanism for normalizing the Context Scores for further computation. Accuracy should decrease as the model generates sentences that further deviate from the topic words, indicating that consistency can still be improved. We use self-attention to determine the attention scores for each topic word. We perform masking for self-attention using three trained linear projections: Query (Q), Key (K), and Value (V). The attention scores are generically calculated as

follows:

$$\textit{Self-Attention(Q,K,V)} = \textit{softmax}\frac{(QK^T)}{(\sqrt{d_k})} \cdot V \quad (7)$$

where d_k is the dimension of key vectors, or, as in our experiment training set containing the topic words.

The multi-head attention mechanism computes multiple attention scores in parallel, allowing the model to attend to different parts of the input sequence simultaneously. The outputs of the attention heads are concatenated and linearly transformed to produce the final output:

$$\textit{Multi-Head(Q,K,V)} = \sum_{i=1,\ldots,n}^{N} \text{Head} : (h_1, \ldots, h_n) \cdot W^O \quad (8)$$

In Eq. 8, *W* is the weights' sum. $head_n = Attention(Q.W^{(Q_i)}, K.W^{(K_i)}, V.W^{(V_i)})$, $Q.W^{(Q_i)}, K.W^{(K_i)}, V.W^{(V_i)}$, and W^O are trained linear projections from the embedded layer. This sub-layer of linear projection is a FeedForward Network (FFN), which consists of two linear transformations with a ReLU activation function in between:

$$FFN(x) = \max(0, X \cdot W_1 + b_1) \cdot + \ldots + (W_n + b_n) \quad (9)$$

The W_1, W_2, b_1 and b_2 are parameters for the feed-forward layer. Also, since the transformer architecture does not have any inherent notion of the position of words in a sequence, positional encoding is inherently added to the input embeddings to provide information about the relative positions of words.

To evaluate the quality of generated sentences, we first calculate the accuracy of the 100 sentences generated for each topic using Eq. 10. For that purpose, an extended list of topic words is manually created, for each topic word which is consistent with the topic domain. Then we check, within the generated 100 sentences for a topic, what percentage contain topic words from the corresponding extended list. Here T_i represents topic *T1* to *T4*.

$$\textit{Topic-Acc}(T_i) = \frac{(Original_{Sentences}\ T_i)}{(Generated_{Sentences}\ T_i\ (100))} \quad (10)$$

The cumulative mean accuracy score is obtained from the *Topicwise_Accuracy* values of all the sentence generation cases.

$$C\bar{M}_{Accuracy} = \frac{\sum_{i=1}^{n} Topic\_Acc(T_i)}{4} \quad (11)$$

The cumulative mean accuracy is obtained for all the generated sentences for topic *T1* to *T4* in terms of simultaneous sentence generation per topic.

5 Results and Analysis

Each topic word acts as a cue for the transformer model to generate 10 sentences with a length of 15 to 25 tokens (words) with a temperature of 0.2. We locally measure the generation accuracy for each generated sentence. The accuracy is calculated using Eq. 10. This accuracy is calculated for all the sentences of each topic. Finally, the cumulative mean of all these topic-wise sentences is computed. In case any generated sentence does not contain any topic word, that sentence is discarded. After the sentence generation on 4 topics, our model achieves a cumulative mean accuracy of **89.54%** in generating sentences, demonstrating the model can generate consistent and fluent sentences from the topic words.

In Table 3, a single generated sentence per topic word is shown from a few topic words already mentioned in Table 1. The generation is also crucial for scrutinizing the sentence styles and semantics produced by the proposed Tokenized Generative Transformer.

5.1 Information Validation Parameters

For comparing the document D_1 consisting of original sentences and D_2 containing generated sentences, we investigate the accuracy and generics of the generated sentences with the actual sentences on the same topics. It has already been shown that in unsupervised and semi-supervised approaches, pre-trained word embeddings are replaceable by contextualized word representations (Peters et al., 2018). We already have masked contextual topic words for higher contextuality representations. These word representations can be better pairwise feature scores for semantic evaluation.

To fully utilize the semantic embedding, we propose a *Perplexity-Similarity Score* to achieve the similarity between the comparing documents to understand the complex similarity or polarity structure of the sentences within. Perplexity scaling is essentially useful for diversified learning, as well as for the evaluation metrics of large corpora. For efficient contrastive selection of sentences with similar perplexities, we define an empirical approach as:

$$Perplexity = P_{O(s)} - P_{G(s)} \quad (12)$$

Index	Topic Word	Generated Sentence Corresponding to Each Topic Word
		Topic T1: Medical Topic; Topicwise Accuracy: 0.92
1.	pandemic	**Pandemic** is a phenomenon when a large number of people are infected with the virus.
2.	epidemic	The **epidemic** is the number of large contaminations that are detected in a given period.
3.	vaccine	Pfizer was the first coronavirus **vaccine**.
		Topic T2: Social Topic; Topicwise Accuracy: 0.89
1.	social	**Social** is a term that describes the social interaction of individuals.
2.	distance	The **distance** between two points on a two-dimensional coordinate is Euclidean.
3.	isolation	WHO reports **isolation** as one of the core reasons for depression.
		Topic T3: Research Topic; Topicwise Accuracy: 0.90
1.	dataset	Many COVID-19 related **datasets**, tools, software are created and shared.
2.	measures	**Measures** is the number of steps that can be taken to achieve the desired outcome.
3.	count	Daily **count** of the COVID-19 cases was at an all-time high in the first quarter of 2020.
		Topic T4: Generic Topic; Topicwise Accuracy: 0.85
1.	approach	The **approach** is a simple, straightforward, and cost-effective way to reduce the cost.
2.	crisis	A setback is not a **crisis**, but scope for analytical examination.
3.	initiatives	All the necessary **initiatives** have been taken to slow down the rate of transmission.

Table 2: Topic-wise information generation. The topic words within generated sentences are marked in distinct **colors** for identification purpose.

P is computed as the perplexity of the comparable sentences to the differences in context scores between the actual information (sentences) $(P_{O(s)})$ and generated information $(P_{G(s)})$.

The comparison criteria for each sentence within D_1 and D_2 is cumulated and determined with each iteration up to *n* by a comparable sentence count θ, concerning the perplexity comparison between the sentences, by deducing the *Hadamard product* (Horn, 1990) of temperature and length of each sentence to be compared one to the same linear dimension.

Once the documents are represented as lists for comparison, we perform the multi-document summarization for encoded tokenized representation of the comparable documents. This is simply done because of optimization and efficiency, to tackle the bottleneck of the system while comparing numerous sentences at each step:

$$E = Comp(\theta^{(D_1 \to D_2 \ldots)}) \times (s(T \to t_1, \ldots, t_n)) \quad (13)$$

The encoding scheme takes the comparability factor of each document while fragmenting the generated sentences $(D_2(s))$ and original sentences $(D_1(s))$ into a possible number of elements of the token set *T*. We compute sentence embeddings from comparisons by multiplying encoded tokens. This combines perplexity with pairwise distances between documents to produce a final semantic similarity score, indicating document affinity.

Given two sets of sentences, S_1 and S_2, at first, these sentences are encoded into a numerical format. Let τ be the tokenization function that maps a sentence *s* to a sequence of tokens $\tau_1, \tau_2, \tau_3, \ldots, \tau_n$. Also, let *E* be the encoding function that embeds by mapping each token τ_i to a high-dimensional vector. The encoded input for any sentence *s* can be represented as a matrix *X* of size $n \times t$, where *n* is the number of tokens and *t* is the dimension of the token embeddings. At this step, if any similar sentences are matched from the perplexity perspective, the comparable resultant sentences are printed with their acquired scores. Formally, the Perplexity-Similarity Score can be proposed:

$$\textit{Per-Sim Score} = E \cdot \tau(D_1(s) \approx D_2(s)) \quad (14)$$

The encoded inputs are passed through learnable parameters from the TGT, such as the adaptive self-attention, multi-head attention, and feedforward layer with positional encoding with normalization.

By computing the pairwise variable distance between the embeddings, we can measure the similarity range between the sentences and gain insights into the underlying structure and content of the text. For a better understanding of per-topic comparison metrics within documents D_1 and D_2 from Perplexity-Similarity Score-driven document

Models	ROUGE		METEOR		BLEU		Per-Sim Score	
	Train	Test	Train	Test	Train	Test	Train	Test
T5-11B (Raffel et al., 2020)	0.75	0.72	0.80	0.78	0.85	0.83	0.85	0.85
LLaMA-13B (Touvron et al., 2023)	0.78	0.76	0.82	0.80	0.85	0.82	0.87	0.86
MPT-7B (Team, 2023)	0.76	0.74	0.81	0.78	0.83	0.81	0.84	0.83
GPT Neo-20B (Black et al., 2021)	0.79	0.77	0.83	0.81	0.87	0.85	0.89	0.89
CoLDA+TGT (Ours)	**0.82**	**0.80**	**0.86**	**0.84**	**0.87**	**0.86**	**0.93**	**0.91**

Table 4: Benchmarking of our proposed framework with transformer-based state-of-the-art language models on the open corpus CORD-19.

comparison for each iteration set, we show the classification report in Table 3.

Topic Groups	Acc.	Prec.	Rec.	F1-Score
T1	0.92	0.90	0.90	0.91
T2	0.89	0.91	0.89	0.89
T3	0.90	0.86	0.85	0.90
T4	0.85	0.85	0.85	0.87
Overall	**0.89**	**0.88**	**0.87**	**0.89**

Table 3: Classification report derived from the identification samples.

5.2 Benchmark Evaluation

Dedicated multi-domain NLP tasks have been performed successfully with several transformer architecture-based models, with state-of-the-art results. We evaluate the performances of a few SoTA models on our test data.

For benchmarking, a few language models with comparable parameter sizes[6] are selected for comparison, as the *Unified Text-to-Text Transformer* (T5-11B) (Raffel et al., 2020), *LLaMA* by Meta (13B parameters) (Touvron et al., 2023), *MosaicML Pretrained Transformer* (MPT) (7B parameters) (Team, 2023), and *GPT Neo* (20B parameters) (Black et al., 2021).

In Table 4, the performance of these models is shown on our test data, alongside our proposed framework's overall attained performance. Our framework maintained a stable outcome across the metrics for both the train and test cases, while outperforming the other compared models. The factual similarity between the generated and demonstrated sentences containing the same topic words is also shown in *Per-Sim Score*, where our framework demonstrates a significant enhancement over the other evaluated models. The GPT Neo scores a narrowly followed score in comparison to our framework, indicating the underlying similar architectures (GPT-based) for both the language models.

The superior performance of our framework indicates its ability to generate sophisticated text while preserving factual accuracy. The contextual topics from Co-LDA enable the capture of nuanced contextual relationships. Our tokenizer-based masking technique aids TGT in producing coherent and semantically consistent sentences. As a result, the system pipeline combining these techniques produces an end-to-end method for enhanced performance on factual text generation tasks, demonstrating both fluency and informative correctness.

6 Conclusion

This paper proposes a framework for generating accurate and relevant information from large corpora of text, combining a contextual topic modeling algorithm (Co-LDA) with a tokenizer-based generative transformer (TGT) network. Co-LDA captures the contextual relationships between topics in a corpus, while TGT harnesses the power of state-of-the-art language models to generate informative sentences based on extracted topic words. Experimental findings overcome the limitations of fixed-length sentence comparisons by considering variable-length sentences in training and test cases. Evaluation using several performance indicators and standard validation metrics ensures an informed assessment of the framework's effectiveness and enables meaningful comparisons with existing gold-standard benchmark datasets. Our approach can be adapted and utilized in other domains, facilitating the extraction and generation of impactful insights from high-volume text data.

[6] **B** mentioned with the models stands for parameters sizes in Billions.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58.

D. M. Blei. 2001. Latent Dirichlet Allocation, Advances in Neural Information Processign Systems. *NIPS'01*.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation.

Roger A Horn. 1990. The hadamard product. In *Proc. Symp. Appl. Math*, volume 40, pages 87–169.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. *arXiv preprint arXiv:2304.03245*.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models. *arXiv preprint arXiv:2305.13661*.

Letian Peng, Zuchao Li, and Hai Zhao. 2023. Contextualized semantic distance between highly overlapped texts. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10913–10931.

Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Publisher: OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language models through news summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255.

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, et al. 2020. Cord-19: The covid-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Xiaobao Wu, Anh Tuan Luu, and Xinshuai Dong. 2022. Mitigating data sparsity for short text topic modeling by topic-semantic contrastive learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2748–2760.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.