Project Report

On Natural Language Generation using LLMs

Submitted in partial fulfillment of the requirement for the degree of

Master of Science in Computer Science

Submitted by

Sudipto Ghosh	**22234747025**
Neeti Wason	**22234747030**

Under the Guidance of

Prof. Vasudha Bhatnagar and Dr. Vikas Kumar

Department of Computer Science, University of Delhi

Department of Computer Science

Faculty of Mathematical Sciences

University of Delhi

Academic Year 2023-2024

Declaration

This is to certify that this project report titled **"On Natural Language Generation using LLMs"** is being submitted to the Department of Computer Science, University of Delhi, New Delhi in partial fulfillment of the requirement for the award of the degree of M.Sc. (Computer Science). It is a record of bonafide work carried out by us under the supervision of Prof. Vasudha Bhatnagar and Dr. Vikas Kumar. The matter embodied in the declaration has not been submitted in part or full to any university or institution for the award of any degree or diploma.

Date: January 31, 2024 **Sudipto Ghosh (22234747025)**

Place: New Delhi **Neeti Wason (22234747030)**

Certificate

This is to certify that this project report titled **"On Natural Language Generation using LLMs"** submitted to the Department of Computer Science, University of Delhi, for the award of the degree of M.Sc. (Computer Science), is a research work carried out by Sudipto Ghosh (22234747025) and Neeti Wason (22234747030) under the supervision of Prof. Vasudha Bhatnagar and Dr. Vikas Kumar. This work has not been submitted in part or full to any other University or Institution for the award of any degree or diploma, to the best of our knowledge.

Prof. Naveen Kumar	**Prof. Vasudha Bhatnagar**	**Dr. Vikas Kumar**
(Head)	(Supervisor)	(Supervisor)

Acknowledgement

We would like to extend our profound gratitude to our supervisor Prof. Vasudha Bhatnagar and Dr. Vikas Kumar for their interest, guidance and valuable suggestions throughout this project. We feel honoured and privileged to work under their constant supervision. Without their advice, constructive ideas, positive attitude, and continuous encouragement, it would not have been possible to make such progress in the designated time frame.

Sudipto Ghosh
(22234747025)

Neeti Wason
(22234747030)

Contents

Chapter 1

Introduction

The landscape of Natural Language Processing (NLP) has undergone a transformative shift in recent years, largely attributed to the advent of advanced language models. Among these, Large Language Models (LLMs) have emerged as powerful tools for understanding and generating human-like text. Built upon deep learning architectures, these models showcase unprecedented capabilities in capturing intricate patterns and nuances of natural language.

Natural Language Processing comprises of two major tasks – Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU involves tasks where the input is in natural language, including semantic parsing, natural language inference, and related endeavors. Conversely, NLG focuses on producing fluent, coherent, and practical language outputs for human consumption. In the domain of language processing, both NLU and NLG are of significant importance as the ability to understand and interact with humans is at the core of developing modern intelligent systems.

Natural Language Generation, a subfield of artificial intelligence, is dedicated to the automatic creation of coherent and contextually relevant text. LLMs, exemplified by OpenAI's GPT-3 and its predecessors, have garnered significant attention for their proficiency in generating human-like responses, answering queries, and even producing creatively nuanced writing. The intricate interplay of neural networks within these models enables them to grasp the complexities of language, generating text often indistinguishable from that produced by humans.

In this project, we experiment with two state-of-the-art LLMs – LLaMA-2, an autoregressive language model introduced by Touvron et al. in [1] and Falcon-LM, introduced by Almazrouei et al. in [2], and try to qualitatively assess their natural language generation capability using neural topic modelling on the given prompts.

1.1 Objective

The primary objective of this research is to analyze the diversity in the language model generations, apply a state-of-the-art topic modelling pipeline on the generated text, compute topic coherence and sentiment scores of the resulting topics and compare the two large language models.

1.2 Methodology

We prompt the two popular large language models viz. LLaMA-2 [1] and Falcon-LM [2] with neutral prompts to look for inherent bias in their generations. We investigate the text generation quality of the language models by computing lexical diversity measures of the generations. We employ BERTopic, introduced by Grootendorst in [3], to perform topic modelling on the generated text with different parameters and compute topic coherence measures. We then analyze the intensity of sentiments in the representative documents of the topics to contrast the characteristics of the generations of the two language models.

1.2.1 Data Generation

We first generate 2000 sequences related to India by selecting four themes viz. Indian Economy, Indian Climate, Indian Infrastructure, and Indian Defence, where we use these themes directly as prompts and generate 500 sequences for each theme. We generate these sequences with token lengths randomly chosen between 180 and 200, with both LLaMA-2 and Falcon-LM. We refer to this dataset as Dataset 1.

We then use the Google Scholar API and the `scholarly` module [4] to scrape the publication titles of research papers using the themes as the search keywords. These titles were then used as prompts for the language models to generate sequences. We refer to the dataset containing these sequences as Dataset 2.

As we use the selected themes directly as prompts for creating Dataset 1, no prompt selection strategy was required. For creating Dataset 2, we consider only those research publication titles which contained at least three words. We then took the first 501 results for each selected theme. These titles were then used as prompts and the resulting sequences were used as-is for our experiments.

Table 1.1 shows the distribution of the datasets. These sequences were subsequently used as input for downstream tasks. More discussion on the topic of data generation has been carried out in Section 3.1.1.

Dataset 1	LLaMA-2		Falcon-LM	
Themes	Count	$< \|T\| >$	Count	$< \|T\| >$
Indian Climate	500	190.27	500	189.90
Indian Defense	500	189.45	500	190.25
Indian Economy	500	189.67	500	190.18
Indian Infrastructure	500	189.86	500	190.11

(a) Same Prompts

Dataset 2	LLaMA-2		Falcon-LM	
Themes	Count	$< \|T\| >$	Count	$< \|T\| >$
Indian Climate	501	190.28	501	190.04
Indian Defense	501	190.23	501	190.54
Indian Economy	480	189.85	480	190.05
Indian Infrastructure	501	189.76	501	189.63

(b) Different Prompts

Table 1.1: Count and average sequence lengths of the sequences generated using language models when prompted with (a) themes only, and (b) research publication titles from Google Scholar matching the themes when used as search keywords

1.2.2 Lexical Diversity

Lexical diversity has been defined in [5] as the range of vocabulary deployed in a text that reflects the capacity to access and retrieve target words from a relatively intact knowledge base i.e. lexicon for the construction of higher linguistic units such as sentences and paragraphs. We compute the various lexical diversity measures to discover the characteristics of the generated text in hope of analyzing the differences between language models.

1.2.3 Topic Modeling

Topic modeling is a natural language processing task which is used to automatically identify topics present in a text corpus. The goal is to uncover hidden thematic structure in a large collection of documents. This can be particularly useful for organizing, understanding, and summarizing large datasets of textual information. Traditional topic modelling techniques include Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). We use, BERTopic, the state-of-the-art neural topic modelling framework, for identifying the topics from the generated text. We compute and report three topic coherence scores on the output of the topic models the generated datasets.

1.2.4 Sentiment Analysis

We prompt the LMs with neutral prompts, and try to detect any inherent bias present in these models by finding the sentiment intensities of the topics extracted from the generated texts. The VADER (Valence Aware Dictionary and Sentiment Reasoner) library, authored by Hutto and Gilbert in [6] is a popular tool for sentiment analysis of the text. Designed for short and informal text, it is particularly effective in social media contexts. It assigns polarity scores (positive, neutral, or negative) to words and calculates an overall compound score for the entire text, indicating sentiment intensity. VADER is sensitive to elements like emoticons, capitalization, and negations, making it suitable for diverse language styles. We use the VADER tool to compute sentiment polarity scores of the representative documents of extracted topics.

1.3 Organization of the Report

The rest of the report is organized as follows: In **Chapter 2** we provide an in-depth analysis of relevant research papers that have informed and influenced the current project. In **Chapter 3**, we present the experiment setup and experiments that were performed to gain insights into the structure and characteristics of the output. In **Chapter 4**, we present the result and analysis of the experiments where we report our findings when using different datasets and language models. In **Chapter 5** we put forward our conclusions and possibilities of future work related to the project.

Appendix A contains the implementation details of experiments and other additional information. **Appendix B** contains details of the project source code and downloadable datasets introduced in the project.

Chapter 2

Background

In the language modelling task, given a sequence of tokens $\mathbf{x} = x_1, \ldots, x_n$, a language model (LM) needs to learn to predict a target token based on the preceding tokens in a sequence. In order to train such a model, the likelihood $\mathcal{L}_{LM}$ is maximized. Post training, the model needs to predict the most likely token x_t^* based on the previously generated tokens $x_1, x_2, \ldots, x_{t-2}, x_{t-1}$. However, sampling based methods can also be used to select the next token based on the distribution to include diversity in the generation. Language models also implement different sampling techniques to inculcate randomness to improve quality viz. temperature sampling, top-k sampling and top-p sampling.

$$\mathcal{L}_{LM}(\mathbf{x}) = \sum_{i=1}^{n} \log P(x_i \,|\, \mathbf{x}_{<i}) \qquad x_t^* = \arg\max_{x_t} P(x_t \,|\, \mathbf{x}_{<t}) \qquad x_t^* \sim P(x_t \,|\, \mathbf{x}_{<t})$$

Statistical Language Models (SLMs) use statistical learning methods to predict the next word based on the most recent context. Neural LMs use neural architectures like multilayer perceptrons and recurrent networks to characterize the distribution of words. Pretrained Language Models (PLMs) involve *pre-training* a neural architecture and then *fine-tuning* the network according to the task. Large Language Models (LLMs) is an umbrella term for PLMs that typically have upwards of billions of trainable parameters, e.g., GPT-3 [7] with 175 billion, PaLM [8] with 540 billion trainable parameters. The idea behind the *large* number of parameters is that the performance of a pretrained LM on downstream tasks is believed to improve on increasing the model or data size [9]

Since the introduction of the Transformer architecture (Fig 2.1) by Vaswani et al. in [10], a novel neural network model for sequence-to-sequence tasks, it has become a foundational model for natural language processing and other sequence-based tasks. The authors achieved state-of-the-art performance with this architecture on various machine translation tasks. Transformer-based models outperform existing models while being

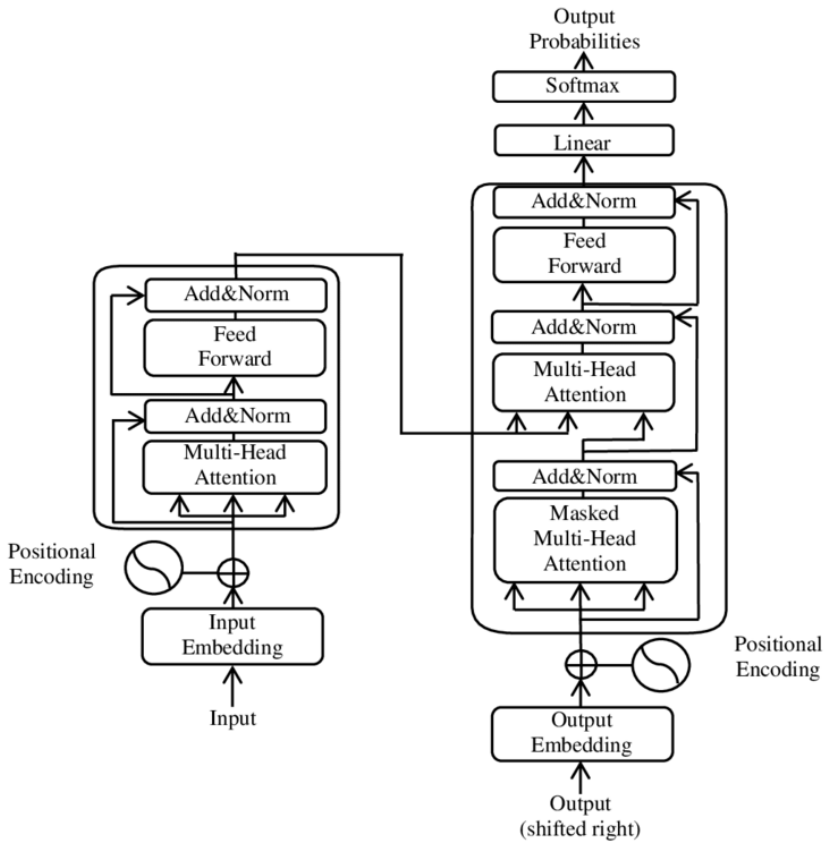


Figure 2.1: Transformer architecture as proposed by Vaswani et al. in [10]

highly parallelizable, facilitating efficient training. Unlike previous approaches that relied on recurrent or convolutional layers, the Transformers employ self-attention mechanisms, enabling the model to consider different parts of the input sequence simultaneously. The attention mechanism [11] allows the model to assign different weights to different positions in the input sequence, capturing long-range dependencies more effectively. This architecture sparked the creation of popular encoder-only language models like BERT [12] and causal decoder-only models like GPT [13].

The LLaMA family of language models introduced by Touvron et al. in [14] and [1] introduce several modifications to the standard Transformers architecture [10] to enhance its performance and capabilities. The model incorporates pre-normalization, which involves normalizing the input of each transformer sub-layer using RMSNorm [15], where activations are rescaled by the root mean square of the summed activations. LLaMA uses the SwiGLU activation [16] – a combination of Swish [17] and GLU [18] activations, instead of the conventional ReLU activation.

$$SwiGLU(x, W, V, b, c, \beta) = Swish_\beta(xW + b) \otimes (xV + c)$$

$$Swish_\beta(x) = x \cdot \text{sigmoid}(\beta x) \quad GLU(a, b) = a \otimes \text{sigmoid}(b)$$

Instead of using absolute positional embeddings, LLaMA models use rotary positional embeddings [19] at each layer. LLaMA was trained on a vast amount of data, specifically 1.4 trillion tokens sourced from publicly available data. The training data had been extended by 40 percent and the context length has been increased from 2000 to 4000 in LLaMA-2 to enhance the contextual understanding of the model. LLaMA models also incorporate the concept of grouped-query attention [20] which contributes to the model's ability to handle diverse queries effectively. These improvements to the Transformers architecture allowed LLaMA-2 models to be competent to recent open-source and proprietary LLMs.

More recently, the Falcon series of causal decoder-only language models proposed by Almazrouei et al. [2] improve upon language models like LLaMA-2 and outperform various SOTA LLMs. These models are based on the PaLM architecture [8] and are pretrained on a dataset of 3500 billion tokens [21]. They propose multi-group attention mechanism, based on multi-query attention [22], to improve the scalability of inference. The models use rotary embeddings [19]. Unlike LLaMA-2, Falcon-LM does not use the GLU activation because of the increased memory footprint. Instead of GLU, they use the Gaussian Error Linear Unit (GeLU) activation. Falcon models also employ parallel attention and MLP blocks and remove biases from linear layers.

$$GeLU(\mathbf{x}) = 0.5x \otimes [1 + erf(\mathbf{x}/\sqrt{2})], \quad erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Large Language Models such as the ones discussed above have shown remarkable performance in many tasks in NLP. However, they sometimes might exhibit unintended behaviours like *hallucination* and generate gibberish or present facts that are falsified or biased [23]. Dhamala et al. in [24] introduce a fairness benchmark dataset of 23679 text generation prompts to investigate biases in open-ended natural language generation from GPT-2 [25], BERT [12] and CTRL [26]. They also present novel metrics for toxicity, psycholinguistic norms and text gender polarity to measure social biases in open-ended text generation. Their results show negative biases of GPT-like models towards certain groups and underline the need to study fairness in NLG to avoid detrimental biases in downstream tasks, where they might snowball and pose larger social problems.

As language expression abilities in humans develop as they grow up [27] [28], they develop better communication skills. When it comes to systems, researchers have been trying to train them to communicate like humans [29]. One of the indicators of development of language understanding in humans is the increase in diversity of words being used in day-to-day communication as a result of an increasing size of vocabulary. Lexical diversity measures, which are also used to identify neural abilities in clinical settings [30], are usually used to measure the quality of sequences [31].

The topic modelling (TM) task is to obtain two sets of distributions from a text corpus – (i) a set T of K distributions over V tokens, where K is the number of topics to be modelled and V is the vocabulary size of the corpus, and (ii) a set Z which comprises of a distribution for each document in corpus over K topics. A topic is essentially a cluster of words taken from a document which are semantically similar. Conventional models like LDA, NMF, and CTM have been used extensively in TM tasks [32] [33], however, they struggle with words that have multiple meanings (polysemy) or different words that convey similar meanings (synonymy). Neural models are better at capturing contextual information better than conventional models. Since the advent of neural topic modelling, which is more flexible and scalable, several approaches [34] have been proposed that cluster word and document embeddings to separate topic modelling from topic representation.

Grootendorst introduces a state-of-the-art neural topic modeling approach in [3] that takes advantage of the language modelling capabilities of transformer-based models and uses a novel class-based TF-IDF procedure to extract topics from document clusters. They generalize the TF-IDF procedure [35] involving term frequency and inverse document frequency to support clusters of documents and model the importance of words in clusters instead of individual documents. The term frequency models the frequency of term t in a class c i.e. the collection of documents concatenated into a single document for each cluster. The inverse document frequency is replaced by the inverse class frequency to measure how much information a term provides to a class. It depends on the logarithm of the ratio of the average number of words per cluster A to the frequency of the term t across all classes. The c-TF-IDF score for a word is thus computed as

$$W_{t,c} = tf_{t,c} \cdot \log\left(1 + \frac{A}{tf_t}\right)$$

They convert documents into dense vector embeddings using a pre-trained language model SBERT [36], and form clusters of these embeddings before performing topic extraction. As clustering algorithms perform poorly on high-dimensional data [37], the dimensionality of the embeddings is reduced to overcome the curse of dimensionality [38]. Well-known techniques for reducing the dimensionality of data like PCA, SVD and t-SNE are supported along with UMAP [39] which is shown to preserve more of local and global features and can be used with language models with different dimensional spaces as it makes no assumption about embedding dimensions. Their approach supports both hard and soft clustering algorithms like k-means, agglomerative clustering and a hierarchical adaptation of DBSCAN. We use the approach in [3] over conventional methods as it has been shown to exhibit higher topic coherence and diversity scores on benchmark datasets like 20 NewsGroups [40] and BBC News [41].

Topic models learn topics from unlabelled documents in an unsupervised fashion. Many researchers [42] [43] have proposed measures to quantify the coherence of a set of facts or statements. When these measures are used in tandem with topic models [44] [45] [46], they can be used to evaluate whether the extracted topics conform to human topic rankings [47]. Topic coherence measures score each topic by measuring the degree of semantic similarity between high-scoring words in the topic, reflecting the human intuition that similar words make up a distinct topic.

Opinions and sentiments of people towards entities can be mined from a corpus using analysis of sentiment words that reflect the emotions regarding a target [48]. Hutto and Gilbert propose a domain-agnostic rule-based model for sentiment analysis in [6] which even outperforms individual human raters and can be used in different contexts. They use a lexicon of 7500 words augmented with sentiment intensity or *valence* scores and rules that capture sentiment intensity and grammatical context. The model is designed to handle negations, booster words, and special characters common in social media text. The compound sentiment polarity (CSP) for a sequence is finally computed as

$$CSP = \frac{\sum_{i=1}^{n} valence_i}{\sqrt{\sum_{i=1}^{n}(valence_i)^2 + \alpha}}$$

As LLMs became popular, there has been an explosion of synthesized texts available on the World Wide Web. Mao et al. in [49] uncover biases towards words that frequently occur in the training data and changes in the way a pre-trained language model has been prompted are observed to change the sentiment in the generation. The works of Sheng et al. in [50] and Dhamala et al. in [24] show that LLM generations show negative bias towards certain social groups and mitigation of biases in NLG becomes very important as generations might eventually end up on the Web and promote stereotypes. Bhatt et al. in [51] have worked to adapt fairness research for the Indian context in NLP tasks based on [52] and discuss various societal and geo-cultural axes that are different from the Western context.

Chapter 3

Experiments

3.1 Experiment Setup

3.1.1 Data Generation

We generate two datasets for our experiments. The first dataset consists of sequences generated on same prompts for four themes – Indian economy, Indian climate, Indian defence and Indian infrastructure – using LLaMA-2 and Falcon-LM. As introduced earlier, the second dataset contains sequences generated with different prompts which comprised of 1983 research publication titles as scraped from Google Scholar. LLaMA-2 took 4718 seconds for generating the first dataset consisting of 2000 sequences, while Falcon-LM took 18363 seconds. On the other hand, LLaMA-2 took 4665 seconds for generating 1983 sequences for the second dataset from the 1983 different prompts, while Falcon-LM took 15869 seconds for the same number of prompts.

Typically, models are trained on high-performance compute servers with higher precision, but quantization allows model representation with lower precision, significantly reducing memory requirements and allows us to perform inference at the edge. However, this trade-off may slightly impact model performance. We initially leverage Hugging Face's `bitsandbytes` library [53], where the model was dynamically loaded with 4-bit quantization, making it feasible for inference and training on platforms with limited resources. Despite memory savings, unexpected outputs emerged during testing, including mixed-code generation and gibberish. This highlighted the delicate balance between memory optimization and preserving model performance, prompting a reassessment of the quantization strategy. We finally performed experiments with the 7-billion parameter versions without any quantization on a Intel ® Xeon ® Gold server with 256 GB RAM and 48 GB VRAM.

The distribution of prompts across themes was shown in Table 1.1. We use the generated sequences as-is and avoid removing stop words as it is not advised for transformer-based embedding models as we need the full context in order to create accurate embeddings with a BERT-based model. CountVectorizer [54] was used for preprocessing after generating embeddings and clustering documents in Section 3.1.2 to remove frequently occurring non-content words.

3.1.2 Topic Modelling

We perform neural topic modelling using BERTopic [3] to extract topics from the sequences. It leverages the `transformers` library and class-based TF-IDF scores to create dense clusters of documents for topic modelling. For each topic, we get a list of topic words with corresponding word scores, count of documents assigned that topic, a topic representation consisting of the topic words and the topic probability. We first encode the sequences for the topic modelling pipeline using the SentenceTransformers framework [36]. We use the `all-mpnet-base-v2` model to generate 768-dimensional embeddings as recommended by the author of [3] due to its better position on HuggingFace's Massive Text Embedding Benchmark (MTEB) Leaderboard [1] and performance on benchmark datasets. The embedding model is a sentence and short paragraph encoder based on [55].

We then use various supported dimensionality reduction techniques and clustering algorithms, which are important parameters for topic modelling with BERTopic, to come up with various topics to analyze further.

Dimensionality Reduction Techniques

1. **Principal Component Analysis**: It is a statistical method whose goal is to transform a dataset into a new coordinate system, where the features are orthogonal, and the most important information is captured by the first few principal components.

2. **Uniform Manifold Approximation and Projection**: Introduced in [39], this technique constructs a high dimensional graph representation of the data then optimizes a low-dimensional graph to be as structurally similar as possible.

3. **Truncated Singular Value Decomposition**: SVD is a known factorization method that decomposes a matrix into three other matrices, representing its singular value decomposition $X = U\Sigma V^T$. In Truncated SVD, the decomposition is *truncated* by keeping only the top k singular values and their corresponding singular vectors, effectively reducing the dimensionality of the data.

[1] https://huggingface.co/spaces/mteb/leaderboard

(a) LLaMA-2 (b) Falcon-LM

Figure 3.1: Wordclouds created from Dataset 1 by the language models

(a) LLaMA-2 (b) Falcom-LM

Figure 3.2: Wordclouds created from Dataset 2 using the language models

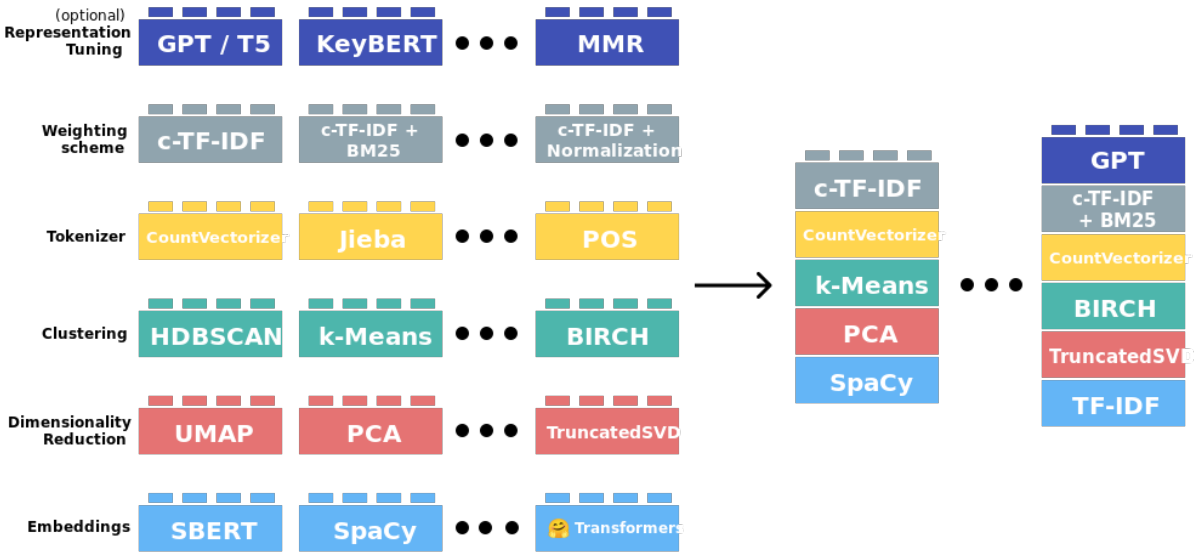


Figure 3.3: Modular architecture of the BERTopic neural topic modelling pipeline

Clustering Techniques

1. *k*-**Means Clustering**: This algorithm partitions a dataset into distinct groups or clusters based on similarity patterns among data points. This method aims to minimize the within-cluster sum of squares, assigning each data point to the cluster whose centroid is closest. The algorithm iteratively refines the cluster assignments until convergence, making it computationally efficient for large datasets.

2. **Agglomerative Clustering**: It is a hierarchical clustering technique that starts by considering each data point as an individual cluster and iteratively merges the closest clusters until only one cluster, encompassing all the data points, remains. The merging of clusters is determined by a linkage criterion that measures the similarity or dissimilarity between clusters. We use the Ward's linkage which minimizes the variance within the merged clusters.

3. **HDBSCAN**: This algorithm was introduced in [56] and is particularly well-suited for discovering clusters of varying shapes and densities in a dataset. It is an extension of the DBSCAN algorithm, which is known for its ability to find clusters based on the density of data points. HDBSCAN builds upon DBSCAN by incorporating a hierarchical approach to cluster discovery. It constructs a tree of clusters using the concept of mutual reachability distance. The algorithm allows for the identification of clusters at different levels of granularity, making it useful for datasets with clusters of varying sizes and shapes.

Topic Representation

After we apply a combination of the dimensionality reduction and clustering techniques, a cluster of documents i.e. a topic is represented as a ranked list of words that are representative of the cluster. Each word is represented by its c-TF-IDF score.

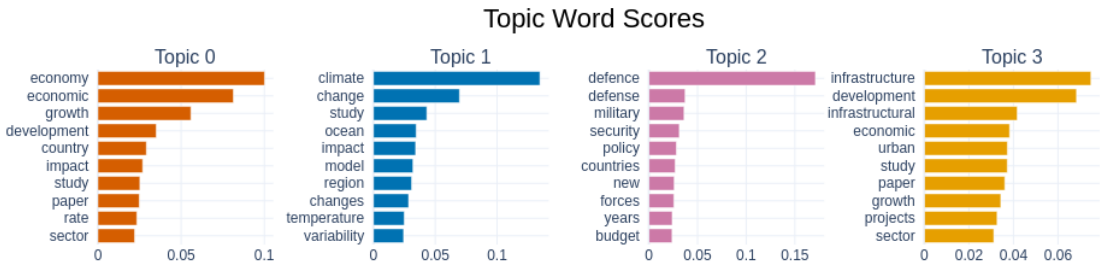


Figure 3.4: Topic and the top ten topic words as output by BERTopic on a dataset

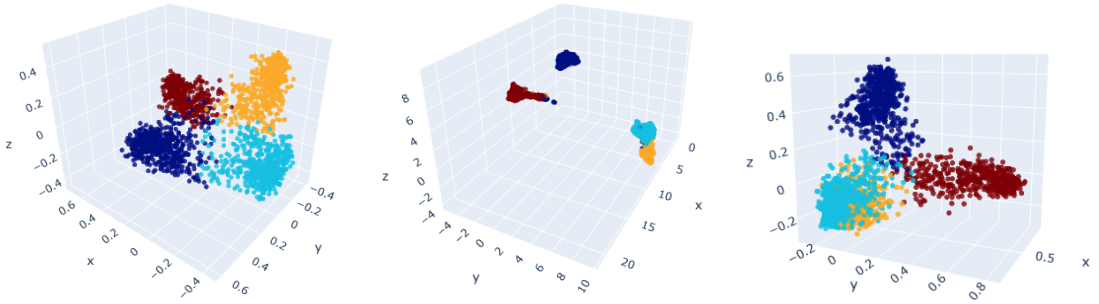
3.1.3 Research Questions

We pose the following research questions before proceeding with the experiments:

1. How diverse are the generations between the two language models?
2. If these generations are modelled into topics, are the topics coherent?
3. What is the sentiment of the models on different themes?

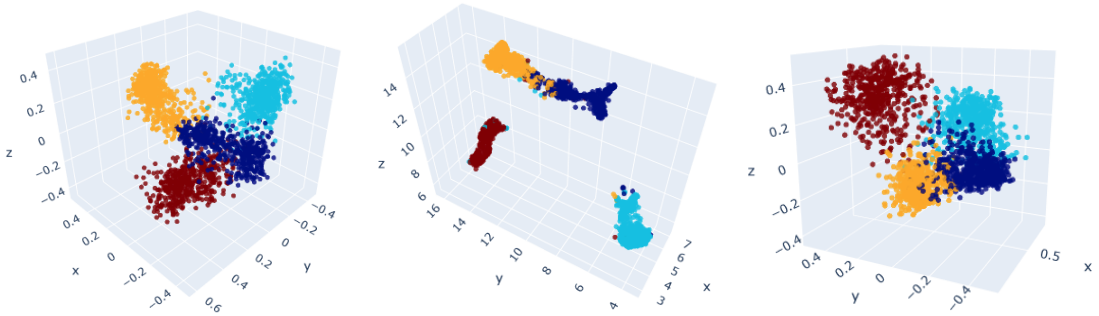
3.2 Computing Diversity in LLM Generations

We lemmatize the generated sequences in each dataset and compute various type-token ratios (TTRs) for comparing lexical diversity between LLaMA-2 and Falcon-LM. Type-Token Ratio (TTR) [31] is computed as the ratio of unique lexical items divided by the total number of words in a sample, and serves as an indicator of lexical diversity in written texts or spoken language, reflecting the range of vocabulary employed. Root TTR or Guiraud's TTR [57] involves taking the the ratio of the number of word types to the square root of word tokens. Log TTR or Herdan's C [57] applies the logarithm function to TTR and is particularly useful when dealing with texts of varying lengths. Maas' TTR [58] is the ratio of the difference of the logarithm of the number of word tokens and the logarithm of the number of token types to the square of the logarithm of the number of tokens. Mean-Segmental TTR [59] involves dividing the text into segments and calculating the TTR for each segment, then averaging these values. The Moving Average



PCA UMAP Truncated SVD

(a) LLaMA-2 Generations



PCA UMAP Truncated SVD

(b) Falcon-LM Generations

Figure 3.5: Document clusters of Dataset 1 identified by BERTopic (PCA/k-Means) visualized in reduced embedding spaces after applying dimensionality reduction techniques

LD Measure	Abbrev.	Formula
Type-Token Ratio	TTR	T/W
Guiraud's TTR	Root TTR	$T/\sqrt{W}$
Herdan's C	Log TTR	$\log T/\log W$
Maas' TTR	MAAS	$(\log W - \log T)/(\log W)^2$
Mean-Segmental TTR	MSTTR	$\sum_{i=1}^{n} TTR(segment_i)/n$
Moving-Average TTR	MATTR	T_{window}/W_{window}

Table 3.1: TTR-based Lexical Diversity Measures computed for the datasets where T is the number of unique token types and W is the number of word tokens

TTR [60] quantifies lexical diversity by computing TTRs in consecutive non-overlapping segments of a given dataset and taking the average of these estimated TTRs.

TTR indices exhibit sensitivity to sequence length due to the likelihood decrease of new words being introduced as the sample length increases [61], however, we decide to keep these indices and compute non-TTR indices of lexical diversity. These include HD-D scores [62] based on the hypergeometric distribution and Measure of Textual Lexical Diversity (MTLD) [62] i.e. the mean length of sequential words in a text that maintain a given TTR value. We list the lexical diversity measures that we consider in Table 3.1.

3.3 Computing Coherence of Topic Models

After performing Topic Modelling on the datasets using BERTopic [3] as described in Section 3.1.2 using all combinations of mentioned dimensionality reduction and clustering techniques, we use the `gensim` [63] library to compute three of the topic coherence scores explored in [44] viz. C_V, C_{UMass} and C_{NPMI}.

The pipeline to compute topic coherence in `gensim` consists of segmentation, probability estimation and taking the arithmetic mean of confirmation measures. A topic coherence metric measures how well the words that make up the topic are supported by the reference corpus. We compute the coherence scores from the top 3 topics when using single themes, and from the top 8 topics when computing for all themes.

C_V [44] employs a sliding window with a size of 110, a segmentation of the top words into a single set, and an indirect confirmation measure utilizing normalized point-wise mutual information (NPMI) and cosine similarity.

$$C_v = \sum_{k=1}^{K} \sum_{n=1}^{N} \frac{\cos(\vec{w}_{n,k}, \vec{w}_k)}{N \cdot K}$$

C_{UMass} [42] relies on document cooccurrence counts, a segmentation based on preceding elements, and a confirmation measure using logarithmic conditional probability.

$$C_{\text{UMass}} = \frac{2}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_i)}$$

C_{NPMI} [43] is a coherence measure that utilizes a sliding window and calculates NPMI for all word pairs among the specified top words.

$$C_{\text{NPMI}} = \frac{2}{N(N-1)} \sum_{i=i}^{N-1} \sum_{j=i+1}^{N} \left(\frac{\log \left(\frac{P(w_i, w_j)+\epsilon}{P(w_i)P(w_j)} \right)}{-\log(P(w_i, w_j) + \epsilon)} \right)^{\gamma}$$

where N is the number of top words in the topic, w_i and w_j are words in the topic, $P(w_i, w_j)$ is the joint probability of words w_i and w_j occurring together, $P(w_i)$ and $P(w_j)$ are the probabilities of words w_i and w_j occurring respectively, ϵ is a smoothing factor, and γ is a parameter that assign more emphasis on context features with high PMI (or NPMI) values with a topic word – for our experiments, we set $\gamma = 2$ as recommended by Aletras et al. in [43].

We also qualitatively look into precision-recall metrics to assess the relevance of the topics. Precision in this case refers to the fraction of relevant documents among the total number of documents assigned to a particular topic. Recall refers to the fraction of relevant documents that have been assigned to a particular topic among the total number of documents actually belonging to a given theme. High recall means that a topic is effectively capturing most of the relevant documents. High precision means that when a topic is assigned to a document, it is likely to be relevant. F1-score is the harmonic mean of the precision and recall and combines both metrics into a single score.

3.4 Sentiment Analysis on Topics

We take the representative sequences of each topic (cluster) and compute sentiment scores using VADER [6]. The representative sequences of a topic are the sequences that contribute to the formation of the cluster, and the topic words are said to be derived from these sequences. We report the normalized VADER compound scores, which take into account both positive and negative sentiment intensities, in Section 4.

Chapter 4

Results

In Table 4.1, we report the lexical diversity measures discussed in Section 3.2 for the datasets of sequences generated on prompts on the themes – (i) Indian Climate, (ii) Indian Defence, (iii) Indian Economy and (iv) Indian Infrastructure. We also consider a dataset with the combination of all sequences and refer to it as (v) All Themes.

When it comes to topic modelling, we have no guarantee on the interpretability of the output of topic models. Therefore, measuring the quality of topics becomes important, more so when neural architectures are used for this task. We use three coherence measures viz. C_V, C_{UMass}, and C_{NPMI}, as described in Section 3.3, to evaluate the quality of the topics. Table 4.2 illustrates the values of these coherence measures when using different parameters for topic modelling with BERTopic.

We report the macro-averaged precision-recall metrics in Table 4.3. Both LLMs achieve a high precision as well as recall with k-Means clustering ($k = 4$). In some cases, documents on the themes *infrastructure* and *economy* get clustered together, which might explain loss of recall.

Finally, we present the compound sentiment scores of the representative sequences of the topics obtained by the topic models with topics with the highest coherence in Table 4.4. We take the average of the top three topics when concerned with datasets on a single theme, and that of the top eight topics when considering a combination of all themes.

Metric		LLaMA-2	Falcon-LM
TTR	i	0.559554	0.594627
	ii	0.636783	0.638263
	iii	0.599143	0.611832
	iv	0.612910	0.640589
	v	0.602097	0.621328
Root TTR	i	6.538296	7.031603
	ii	6.800674	7.119058
	iii	6.841200	7.030286
	iv	6.974624	7.166018
	v	6.788698	7.086741
Log TTR	i	0.881118	0.894201
	ii	0.904111	0.906615
	iii	0.894397	0.899077
	iv	0.898939	0.907461
	v	0.894641	0.901839
MAAS	i	0.055696	0.049173
	ii	0.046392	0.044221
	iii	0.049711	0.047207
	iv	0.047731	0.043737
	v	0.049882	0.046085
MSTTR	i	0.832140	0.862199
	ii	0.883602	0.890433
	iii	0.869656	0.879628
	iv	0.874087	0.890693
	v	0.864871	0.880738
MATTR	i	0.832076	0.862297
	ii	0.881899	0.884411
	iii	0.873268	0.880359
	iv	0.873692	0.885652
	v	0.865234	0.878180
HD-D	i	0.751589	0.770547
	ii	0.798909	0.796046
	iii	0.786984	0.778959
	iv	0.788224	0.801840
	v	0.781427	0.786848
MTLD	i	49.155613	62.653386
	ii	70.202863	74.724666
	iii	62.823810	68.079342
	iv	65.249444	75.111651
	v	61.857932	70.142261

(a) Same Prompts

Metric		LLaMA-2	Falcon-LM
TTR	i	0.590563	0.595385
	ii	0.590145	0.611065
	iii	0.565967	0.604452
	iv	0.583385	0.592762
	v	0.582690	0.600879
Root TTR	i	7.007476	7.155216
	ii	6.984259	7.031694
	iii	6.771099	6.666449
	iv	6.941172	6.966247
	v	6.927642	6.957956
Log TTR	i	0.892987	0.895250
	ii	0.892696	0.898516
	iii	0.884628	0.894857
	iv	0.890574	0.893476
	v	0.890280	0.895532
MAAS	i	0.049728	0.048269
	ii	0.049993	0.047722
	iii	0.053485	0.049839
	iv	0.050873	0.049475
	v	0.050994	0.048815
MSTTR	i	0.858986	0.862673
	ii	0.857717	0.857300
	iii	0.838526	0.849814
	iv	0.856497	0.859429
	v	0.853084	0.857383
MATTR	i	0.855897	0.858050
	ii	0.857009	0.855074
	iii	0.841270	0.850891
	iv	0.855059	0.858115
	v	0.852426	0.855582
HD-D	i	0.772965	0.767063
	ii	0.758233	0.746862
	iii	0.740491	0.711976
	iv	0.775992	0.750161
	v	0.762147	0.744355
MTLD	i	60.408203	62.611663
	ii	59.251547	62.045965
	iii	52.530323	54.782784
	iv	58.417324	59.396287
	v	57.706087	59.761345

(b) Different Prompts

Table 4.1: LD Scores using (a) Same and (b) Different Prompts for (i) Indian Climate, (ii) Indian Defence, (iii) Indian Economy, (iv) Indian Infrastructure and (v) All Themes

Metrics	Expt.	LLaMA-2		Falcon-LM	
C_V	i	0.7986	SVD/HDBSCAN	0.7074	PCA/HDBSCAN
	ii	0.6441	SVD/HDBSCAN	0.6377	SVD/HDBSCAN
	iii	0.5562	PCA/HDBSCAN	0.5589	UMAP/HDBSCAN
	iv	0.5129	SVD/Agglo.	0.6028	PCA/HDBSCAN
	v	0.8494	UMAP/HDBSCAN	0.7930	PCA/KMeans
C_{UMass}	i	-0.0824	SVD/Agglo.	-0.0757	SVD/Agglo.
	ii	0.0000	SVD/KMeans	-0.0355	SVD/KMeans
	iii	0.0000	SVD/KMeans	-0.0328	PCA/KMeans
	iv	0.0000	UMAP/KMeans	0.0000	PCA/Agglo.
	v	-0.1120	PCA/KMeans	-0.0315	UMAP/HDBSCAN
C_{NPMI}	i	0.2499	SVD/HDBSCAN	0.1930	PCA/HDBSCAN
	ii	0.1684	SVD/HDBSCAN	0.1655	PCA/HDBSCAN
	iii	0.1316	PCA/HDBSCAN	0.0842	PCA/HDBSCAN
	iv	0.0847	SVD/Agglo.	0.0964	UMAP/HDBSCAN
	v	0.2791	PCA/HDBSCAN	0.2112	PCA/HDBSCAN

(a) Same Prompts

Metrics	Expt.	LLaMA-2		Falcon-LM	
C_V	i	0.6272	PCA/HDBSCAN	0.6601	SVD/HDBSCAN
	ii	0.6567	UMAP/HDBSCAN	0.5685	UMAP/HDBSCAN
	iii	0.6229	UMAP/HDBSCAN	0.6126	SVD/HDBSCAN
	iv	0.6805	PCA/HDBSCAN	0.5538	PCA/HDBSCAN
	v	0.6878	PCA/HDBSCAN	0.7219	SVD/Agglo.
C_{UMass}	i	-0.0426	PCA/KMeans	-0.0385	SVD/Agglo.
	ii	-0.0509	PCA/Agglo.	-0.0213	SVD/Agglo.
	iii	-0.0928	SVD/Agglo.	-0.0294	PCA/Agglo.
	iv	-0.0170	PCA/Agglo.	-0.0581	PCA/Agglo.
	v	-0.0145	UMAP/Agglo.	-0.0360	SVD/HDBSCAN
C_{NPMI}	i	0.1100	SVD/HDBSCAN	0.1885	PCA/HDBSCAN
	ii	0.0523	UMAP/Agglo.	0.0877	PCA/HDBSCAN
	iii	0.1040	PCA/HDBSCAN	0.1367	SVD/HDBSCAN
	iv	0.1874	PCA/HDBSCAN	0.0952	PCA/HDBSCAN
	v	0.1675	PCA/HDBSCAN	0.1590	SVD/Agglo.

(b) Different Prompts

Table 4.2: Best values of the coherence measures of topics extracted from sequences generated by LLaMA-2 and Falcon-LM using (a) Same Prompts and (b) Different Prompts (taken from Google Scholar articles) for different themes – (i) Indian Climate, (ii) Indian Defence, (iii) Indian Economy, (iv) Indian Infrastructure and (v) All Themes

Metric	LLaMA-2	Falcon-LM
Precision	0.97	0.97
Recall	0.97	0.97
F1 Score	0.97	0.97

(a) Same Prompts

Metric	LLaMA-2	Falcon-LM
Precision	0.93	0.92
Recall	0.93	0.91
F1 Score	0.93	0.91

(b) Different Prompts

Table 4.3: Macro-averaged Precision, Recall and F1-Score of topic assignment done on sequences generated from (a) Same Prompts (Best with UMAP/KMeans) and (b) Different Prompts (Best with PCA/KMeans)

Theme	LLaMA-2	Falcon-LM
Indian Climate	0.00128	0.40141
Indian Defence	0.59435	0.21567
Indian Economy	0.76258	0.31241
Indian Infrastructure	0.77870	0.70843
All Themes	0.29037	0.17534

(a) Same Prompts

Theme	LLaMA-2	Falcon-LM
Indian Climate	0.65183	0.31178
Indian Defence	0.55903	0.74556
Indian Economy	0.74089	0.63033
Indian Infrastructure	0.63210	0.56170
All Themes	0.92433	0.80773

(b) Different Prompts

Table 4.4: Averaged compound sentiment polarity scores obtained for documents belonging to topics extracted by topic modelling for different themes using generations for (a) Same Prompts and (b) Different Prompts

Chapter 5

Conclusion and Future Work

In this project, analyze the diversity in the text generated by two LLMs – LLaMA-2 and Falcon-LM, apply topic modelling on the generated text, and compute topic coherence scores and sentiment polarity scores of the resulting topics to compare the models.

Falcon-LM has a higher type-token ratios (TTR, Root TTR, Log TTR, MSTTR, MATTR) and MTLD scores than LLaMA-2 on both datasets which implies an overall higher degree of lexical variation in the model. On the other hand, LLaMA-2 has higher values of MAAS on both datasets as compared to Falcon-LM. When it comes to the HD-D scores, LLaMA-2 has higher scores than Falcon on Dataset 1 while Falcon has higher scores on Dataset 2.

On comparing the topic coherence measures, LLaMA-2 has higher topic coherence than Falcon-LM for all the three measures computed on Dataset 1. When it comes to Dataset 2, we observe LLaMA-2 having higher C_V coherence and Falcon-LM having higher C_{UMass} and C_{NPMI} scores.

When presented with a combined dataset of sequences related to tall themes, the average sentiment scores are higher when different prompts are used for both language models as compared to prompting with the same prompt. LLaMA-2 has a higher sentiment score than Falcon-LM on both datasets.

One can also look into other topic diversity measures as described by Bianchi et al. in [64] instead of lexical diversity measures of the sequences to come to a conclusion about the differences between language models. In place of other topic modelling methods like CTM [64] and Top2Vec [65] can also be utilized instead of BERTopic and the correlation among results of different topic models may be of interest. It may also be useful to use correlate results from bias detection datasets like ToxiGen [66] and BOLD [24]. It may also be interesting to curate similar datasets of other countries and contrast the results.

References

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "LLaMA 2: Open Foundation and Fine-Tuned Chat Models," *arXiv preprint 2307.09288*, 2023.

[2] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, *et al.*, "The FALCON Series of Open Language Models," *arXiv preprint 2311.16867*, 2023.

[3] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," *arXiv preprint 2203.05794*, 2022.

[4] S. A. Cholewiak, P. Ipeirotis, V. Silva, and A. Kannawadi, "SCHOLARLY: Simple access to Google Scholar authors and citation using Python," 2021.

[5] G. Fergadiotis and H. Wright, "Lexical diversity for adults with and without aphasia across discourse elicitation tasks," *Aphasiology*, vol. 25, pp. 1414–1430, 11 2011.

[6] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, pp. 216–225, May 2014.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, "PALM: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[9] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[15] B. Zhang and R. Sennrich, "Root Mean Square Layer Normalization," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[16] N. Shazeer, "GLU variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[17] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[18] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, pp. 933–941, PMLR, 2017.

[19] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with Rotary Position Embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[20] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints," *arXiv preprint arXiv:2305.13245*, 2023.

[21] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay, "The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only," *arXiv preprint arXiv:2306.01116*, 2023.

[22] N. Shazeer, "Fast transformer decoding: One write-head is all you need," *arXiv preprint arXiv:1911.02150*, 2019.

[23] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 27730–27744, Curran Associates, Inc., 2022.

[24] J. Dhamala, T. Sun, V. Kumar, S. Krishna, Y. Pruksachatkun, K.-W. Chang, and R. Gupta, "BOLD: Dataset and metrics for measuring biases in open-ended language generation," in *FAccT*, 2021.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019.

[26] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A Conditional Transformer Language Model for Controllable Generation," *arXiv preprint arXiv:1909.05858*, 2019.

[27] S. Pinker, *The Language Instinct: How the Mind Creates Language*. Penguin UK, 2003.

[28] M. D. Hauser, N. Chomsky, and W. T. Fitch, "The faculty of language: what is it, who has it, and how did it evolve?," *science*, vol. 298, no. 5598, pp. 1569–1579, 2002.

[29] A. M. Turing, *Computing Machinery and Intelligence*. Springer, 2009.

[30] G. Fergadiotis, H. H. Wright, and T. M. West, "Measuring lexical diversity in narrative discourse of people with aphasia," 2013.

[31] J. W. Chotlos, "A statistical and comparative analysis of individual written language samples," *Psychological Monographs*, vol. 56, no. 2, p. 75, 1944.

[32] A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat, and A. Hassan, "Topic Modeling Algorithms and Applications: A Survey," *Information Systems*, vol. 112, p. 102131, 2023.

[33] Y. Chen, H. Zhang, R. Liu, Z. Ye, and J. Lin, "Experimental explorations on short text topic mining between lda and nmf based schemes," *Knowledge-Based Systems*, vol. 163, pp. 1–13, 2019.

[34] S. Sia, A. Dalmia, and S. J. Mielke, "Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too!," *arXiv preprint*

arXiv:2004.14914, 2020.

[35] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," 12 2001.

[36] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[37] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New directions in statistical physics: econophysics, bioinformatics, and pattern recognition*, pp. 273–309, Springer, 2004.

[38] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[39] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform Manifold Approximation and Projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[40] T. Mitchell, "Twenty Newsgroups." UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C5C323.

[41] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, (New York, NY, USA), p. 377–384, Association for Computing Machinery, 2006.

[42] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (R. Barzilay and M. Johnson, eds.), (Edinburgh, Scotland, UK.), pp. 262–272, Association for Computational Linguistics, July 2011.

[43] N. Aletras and M. Stevenson, "Evaluating topic coherence using distributional semantics," in *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)* (A. Koller and K. Erk, eds.), (Potsdam, Germany), pp. 13–22, Association for Computational Linguistics, Mar. 2013.

[44] M. Röder, A. Both, and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, (New York, NY, USA), p. 399–408, Association for Computing Machinery, 2015.

[45] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring topic coherence over many models and many topics," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, (USA), p. 952–961, Association for Computational Linguistics, 2012.

[46] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pp. 100–108, 2010.

[47] J. Chang, S. Gerrish, C. Wang, J. Boyd-graber, and D. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in Neural Information Processing Systems* (Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, eds.), vol. 22, Curran Associates, Inc., 2009.

[48] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2020.

[49] R. Mao, Q. Liu, K. He, W. Li, and E. Cambria, "The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection," *IEEE Transactions on Affective Computing*, 2022.

[50] E. Sheng, K.-W. Chang, P. Natarajan, and N. Peng, "The Woman Worked as a Babysitter: On Biases in Language Generation," 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)* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), (Hong Kong, China), pp. 3407–3412, Association for Computational Linguistics, Nov. 2019.

[51] S. Bhatt, S. Dev, P. Talukdar, S. Dave, and V. Prabhakaran, "Re-contextualizing Fairness in NLP: The Case of India," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Vol. 1: Long Papers)* (Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, eds.), (Online only), pp. 727–740, Association for Computational Linguistics, Nov. 2022.

[52] N. Sambasivan, E. Arnesen, B. Hutchinson, T. Doshi, and V. Prabhakaran, "Re-imagining Algorithmic Fairness in India and Beyond," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, (New York, NY, USA), p. 315–328, Association for Computing Machinery, 2021.

[53] T. Dettmers, M. Lewis, S. Shleifer, and L. Zettlemoyer, "8-bit Optimizers via Block-wise Quantization," *9th International Conference on Learning Representations, ICLR*, 2022.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[55] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MPNet: Masked and Permuted Pre-training for Language Understanding," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, (Red Hook, NY, USA), Curran Associates Inc., 2020.

[56] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining* (J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, eds.), (Berlin, Heidelberg), pp. 160–172, Springer Berlin Heidelberg, 2013.

[57] F. J. Tweedie and R. H. Baayen, "How variable may a constant be? measures of lexical richness in perspective," *Computers and the Humanities*, vol. 32, pp. 323–352, 1998.

[58] H.-D. Maas, "Über den zusammenhang zwischen wortschatzumfang und länge eines textes," *Zeitschrift für Literaturwissenschaft und Linguistik*, vol. 2, no. 8, p. 73, 1972.

[59] D. Malvern, B. Richards, N. Chipere, and P. Durán, *Lexical diversity and language development*. Springer, 2004.

[60] M. Covington, "CASPR research report 2007-05: MATTR user manual," *Athens, GA: The University of Georgia*, 2007.

[61] H. S. Heaps, *Information retrieval: Computational and theoretical aspects*. Academic Press, Inc., 1978.

[62] P. M. McCarthy and S. Jarvis, "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment," *Behavior Research Methods*, vol. 42, pp. 381–392, 2010.

[63] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010.

[64] F. Bianchi, S. Terragni, and D. Hovy, "Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Vol. 2: Short Papers)*, (Online), pp. 759–766, Association for Computational Linguistics, Aug. 2021.

[65] D. Angelov, "Top2Vec: Distributed Representations of Topics," *arXiv preprint arXiv:2008.09470*, 2020.

[66] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar, "ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Dublin, Ireland), pp. 3309–3326, Association for Computational Linguistics, May 2022.

Appendix A

Additional Details

Sequence Generation

These parameters were common for generating sequences using both LMs.

```
SEED = 42
ITERS = 1000
MIN_TOKENS = 180
MAX_TOKENS = 200
MAX_LEN = random.randint(MIN_TOKENS, MAX_TOKENS)
no_repeat_ngram_size = 2
do_sample = True
top_k = 40
top_p = 0.9
temperature = 0.6
```

Lexical Diversity Measures

`window_length = 25` was used for computing MSTTR and MATTR.

Topic Modelling

Dimensionality reduction techniques were initialized as follows:

```
PCA()
UMAP(n_neighbors=180, n_components=50, metric='cosine')
TruncatedSVD(n_components=100, random_state=42)
```

Theme	Prompt
Indian climate	the impact of climate change on indian agriculture
Indian defence	india–japan defence ties: building a strategic partnership
Indian economy	an econometric model of india: estimating prices, their role and sources of change
Indian infrastructure	impact of jnnurm and uidssmt/ihsdp programmes on infrastructure and governance outcomes in cities/towns in india

Table A.1: Sample Research Publication Titles from Google Scholar

Clustering algorithms were initialized as follows:

```
KMeans(n_clusters=4)
AgglomerativeClustering(n_clusters=4, method='ward')
HDBSCAN(min_cluster_size=5, prediction_data=True)
```

BERTopic was initialized as follows:

```
BERTopic(
# Sub-models
embedding_model=SentenceTransformer("all-mpnet-base-v2"),
umap_model=umap_model,  # dimensionality reduction
hdbscan_model=hdbscan_model,  # clustering algorithm
representation_model={
"KeyBERT":  KeyBERTInspired(),
"MMR":  MaximalMarginalRelevance(diversity=0.3),
},
vectorizer_model=CountVectorizer(stop_words="english"),
calculate_probabilities=True,
# Hyperparameters
top_n_words=10,
verbose=True
)
```

Theme	Text
Indian climate	"Indian climate is known for its hot and humid weather. It is a common sight to see people sweating profusely. Apart from the hot weather, the dust, pollution, and other factors add to the problem. The sweat on the skin is not only uncomfortable but also causes many problems. Sweating is the body's natural way of regulating the temperature. However, when it is excessive, it can cause many health problems such as skin infections, fungal infection, rashes, etc. The skin plays a vital role in regulation of body temperature and it also acts as a protective barrier against the environment. When the sweats are excessively produced, this barricade is compromised and the person is left with no option but to deal with the problems that come along with it. One of the common problems caused by sweaty skin are the rushes. These are red patches that ..."
Indian defence	"Indian defence minister Rajnath Singh on Tuesday said that the government is working on a plan to bring in a bill in the winter session of Parliament to amend the Armed Forces Special Powers Act (AFSPA). The AFSP Act which grants special powers to the armed forces in disturbed areas has been in force in Jammu and Kashmir since 1990. The law is also in place in parts of Arunachal Pradesh Manipur Nagaland and Assam. ""We are working to introduce a law in Parliament in winter to replace the AFPSA with a new law. We will try to remove the provisions that are not liked by the people"" Singh said. ""The government will bring a separate law for Jamu and kashmr but it will also apply to other parts where the law ..."
Indian economy	"an indian economic miracle? India is often referred to as a rising star in the global economy. In 2008, the country was the fourth largest economy in Asia and the eleventh largest in terms of nominal GDP. Its economy has been growing at a rate of around 7% for the past decade and it is estimated that it will grow at an average of 8% in 1998-2101. In the early 90s, India's economy was in crisis. The country' s GNP was declining, and its currency was losing value. This led to a series of reforms, which were designed to improve the economy and make it more competitive. The economic situation in India is not as good as it seems. According to the World Bank, in March 30, 09, it was estimated at 5.6%. This ..."

Table A.2: Sample text generated using LLaMA-2

Theme	Text
Indian defence	"the russo-japanese war and the defence of india and british east asia Download Book The Russo Japan War And The Defence Of India And British East Asia in PDF format. You can Read Online The Russo Japan Wars: 1856-1896 and 1894-1991, (2 Vols.), Vol.1 The Japanese - The Russian by John R. Hill File description The Japanese and War Defence The 2 RusssioJapan 1996, of 1860-1864 1885-1914): The in and The The East 1 3 Volumes.The 4 The of The War 5 The Japan The and of and Defence RusxoJapan East East in Defence 8 9 10 The Vol 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 34 36 37 "
Indian economy	"Indian economy is one of the top five economic power houses of 2020. It has been the fastest-growing economy in 2008-2021, with a growth rate of nearly 8.4%. The economy grew by more than 7% every single year between 2014 and 19. According to IMF, India is expected to be among the 3 largest economies in the world after the United States and China (by 2040), overtaking the British economy. The economic growth is a result of several structural and regulatory changes in India in recent years - liberalization of FDI, opening up of key sectors like insurance, banking, telecom, defence – and the resultant increase in investment and productivity. These changes have had a direct impact on the manufacturing sector and its growth. This has also led to an increase demand for skilled labour, leading to increased investment in R&D and training and development"
Indian infrastructure	"Indian infrastructure company Larsen & Toubro Limited (L&T) has bagged an order worth Rs 4,800 crore from the Delhi Metro from which it would be building the 25.5 km Pink Line in the National Capital Region. The project, which will cost Rs4.2 billion and will be completed by 2014 as per a statement by Delhi Chief Minister Sheila Dikhatra. The Pink line will connect Dwarka in West Delhi to the new Badarpur station on the Yellow Line. As per the plans of DDA the Dwakarkar station, the first metro station to be built under Phase Three will come up at Dwaka in 2009 and the second one — at Badri in East Delhi – will begin construction in January 2020 According to a report from The Indian Express, Larsens Managing Director and Chief Executive Officer A K Venky said the ..."

Table A.3: Sample text generated using Falcon-LM

Appendix B

Source Code

The datasets and the source code of the project are available on GitHub [1].

The repository has been organised as follows:

- **data/** contains the datasets for the experiments
  - **[LM]-7b-200/** contains the sequences generated with a LM using same prompts
  - **[LM]-7b-200-google-scholar/** contains the sequences generated with a LM using different prompts
  - **prompts/** contains the different prompts scraped from Google Scholar
- **results/** contains the various measures computed to characterize the datasets
- **notebooks/** contains the IPython notebooks with the source code and subroutines for various pipeline tasks

[1] https://github.com/sudiptog81/llm-nlg-project