# Utility of Large Language Models to Extract Commonsense Knowledge

by

Anika Ahmed
21101029
Nafis Chowdhury
21101034
Moinul Haque
21101186

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

Anika Ahmed
21101029

Nafis Chowdhury
21101034

Moinul Haque
21101186

# Approval

The thesis/project titled "Utility of Large Language Models to Extract Commonsense Knowledge" submitted by

1. Anika Ahmed (21101029)
2. Nafis Chowdhury (21101034)
3. Moinul Haque (21101186)

Of Summer 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October, 2024.

**Examining Committee:**

Supervisor: (Member)

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator: (Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department: (Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

# Abstract

Large Language models are artificial intelligence models that hold the capability to understand and generate natural language text as they are trained using large amounts of data for a lot of languages. The sources these models are trained on include books, articles, websites, and many more. As the large language models know the languages along with their syntax and structures thoroughly, we can expect them to work well for the Bengali language and compose enough knowledge related to the Bengali culture. One of the challenges of working with the Bengali language is the lack of Natural Language Processing methods such as Semantic Parsing, Parts of Speech tagging, and Named Entity Recognition. Our motive was to test the effectiveness of large language models in answering Bengali culture and language-based queries, alongside analyzing which fields of knowledge require improvement. As we do not need Natural Language Processing tools while working with large language models, these models could serve our purpose. Therefore, through our research, we formed a corpus to analyze the utility of large language models for the Bengali language. This corpus aided us in recognizing the gaps of the large language models in terms of factual and cultural commonsense knowledge through natural language processing tasks such as question-answering and masked prediction.

**Keywords:** Natural Language Processing; Large Language Models; Bengali language

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

Any Large Language Model (LLM) is considered to consist of knowledge of any language on a large scale since it is already pre-trained on a huge corpus for several languages. The Bengali language is considered a low-resource language since there is a lack of large-scale annotated data, corpus, and language tools when compared to widely used languages such as English. This is because the language lacks robust natural language processing toolkits such as tokenizers and Parts-of-Speech taggers. Since the large language models are pre trained with huge datasets, they are accustomed to the language. However, the large language models have varying understandings of the Bengali language. Commonsense knowledge can vary from one culture to another. We can also say that generally cultural knowledge is something that can be known as commonsense knowledge. The question "What did you have for lunch?" can have varying answers depending on the culture it is asked in. However, it is expected for us to know which dishes a person living in our surroundings might have for lunch. For example, a Bangladeshi person usually has plain rice for lunch alongside different types of curries whereas an Italian person might be having pasta for their usual lunch. Hence, in order to determine the performance of large language models for Bengali knowledge, we developed a corpus consisting of Bengali terms, context, questions and the popularity of the terms to evaluate the familiarity and effectiveness of large language models towards Bengali cultural, factual and language-based queries. With the help of our corpus, we determined how much of that knowledge already exists in the large language models. In the corpus, there are specific natural language processing tasks like question answering and masked filling to evaluate the knowledge gap of large language models. In our analysis, we analyzed whether the popularity of a topic affects the accuracy of the models in answering questions related to that topic. This portion of the research is to find out whether Retrieval Augmented Generation is required by the large language models to answer questions in Bengali. The performance of monolingual and multilingual models is further analyzed via masked prediction. In our work we tried to find out the categories of Bengali culture regarding which the large language models lack knowledge by the comparison between models and among factual and cultural categories.

## 1.1 Research Statement

In this research, we collected data by web scraping to create a corpus for the Bengali language which will be used to evaluate the performance of large language models.

As methods like Semantic Parsing, Parts of Speech tagging, Named Entity Recognition, etc are not well performing for the Bengali language, and we expect large language models to compose enough Bengali cultural knowledge, we will be testing these large language models to find out the areas where they need improvement. Using large language models does not require such natural language processing toolkits, so we created a corpus through data collection which consists of questions, context, and terms. This corpus was used to test the knowledge gaps within large language models for the Bengali language through masked prediction and question answering. During masked prediction, we used SahajBERT and BanglaBERT which are monolingual models, and the multilingual models Gemini and Llama. For question answering, we did a comparison between the large language models Gemini and Llama.

## 1.2 Research Objective

- We wanted to assess whether the large language models consist of enough cultural and factual Bengali knowledge.
- We created a Bengali dataset through web scraping.
- The collected data was used to perform masked prediction using two monolingual and two multilingual large language models.
- Question answering was done both without context and with context using two large language models.
- The utility of the large language models was determined by finding out the knowledge gap using appropriate metrics and benchmarks after dividing the data into separate categories.

## 1.3 Structure of Report

This report will have the following chapters:

**Chapter 1:** Introduction where we discussed the motivation behind this work, stated the statement of our research and the objective that we will seek to achieve throughout our work.

**Chapter 2:** Literature review will include the method of selecting previous research papers similar to our research and a summary of some papers are included.

**Chapter 3:** Workplan will show our detailed workplan of our work which is followed by a flowchart of works.

**Chapter 4:** Data presents the dataset we created and the steps we took to form it.

**Chapter 5:** Model talks about the models we used to decide the steps we took and the algorithms we used in our data analysis.

**Chapter 6:** In Results and Analysis, we analyzed our outcomes and compared the results given by the large language models.

**Chapter 7:** Conclusion where we described our reason for doing this work and summarized our upcoming work.

# Chapter 2

# Literature Review

## 2.1 Survey Methodology

When we selected our research topic, we sought guidance from our supervisor, who helped us collaborate with an expert in the natural language processing field to work under his supervision. The expert provided documents and research papers related to large language models, and with our supervisor's validation, we researched our literature review with verified papers relating to our topic. We carefully picked and organized our selection of research papers, specifically focusing on large language models, commonsense reasoning, and natural language processing since it was relevant to our research topic. We ensured that papers with substantial citations were included in our literature review. For instance, we primarily focused on papers like Open Mind Common Sense by Singh et al. (2002) [1], CAMeL by Naous et al. (2023) [19], and POPQA by Mallen et al. (2022) [16], as they were very important to our thesis. Most of these papers were published in recent times. We maintained regular communication with our supervisor, who advised a structured approach to reading and summarizing papers, encouraging us to seek clarification on unfamiliar terms and concepts. This proactive engagement and healthy learning environment provided us with a clear understanding and the concept we are studying.

## 2.2 Related Works

We primarily focused our literature review on the problems they are trying to solve, the method of creating datasets, the type of data the datasets consist of, and the analysis they got. We have arranged our selected papers in ascending order based on their publication dates.

In his paper, Singh (2002) [1] talked about The Open Mind Common Sense project,

whose motive was to be able to create a dataset without the help of experts that will be strong enough to be able to give commonsense knowledge to machine learning models. This project focused on taking input from thousands of non-expert volunteers on the internet to increase both the diversity and amount of data points. To construct the dataset, they created a variety of activities that were presented to the volunteers. For example, images were presented to the participants and they were asked to describe the images using natural language. They were also given story titles to write short stories on those topics. The volunteers were encouraged to enter data such that 'even a child could understand' them. Along with these forms of inputs, some of the activities consisted of templates such as 'fill in the blank'. These activities restricted the users from entering knowledge from narrow fields. This was done because these types of data are easier to parse than regular data. Both of these types of activities were used continuously to gather data for the dataset. Both sentence-level inputs and larger structured inputs were taken. After taking inputs from the users, reformulators were used to modify them by paraphrasing, disambiguating, inference, etc. The purpose of modifying the inputs was to supply the machine with multiple and detailed views of each type of knowledge. The dataset consists of sentence-level and story-level data. About 4000 such data points were collected from around 8000 volunteers. The data points were manually decomposed into 90 groups such as 'Story events', 'Grammatical', 'Photo descriptions', etc. However, one-third of the data did not fall into any of the categories. They are working on making more categories to accommodate the rest of the data. While researching they found out that using a controlled subset of English such as templates makes it far easier for people to supply knowledge to systems in a way that is still computer-processable. They also found out that using reformulators for inference is a good method because tracking the inference trace can lead to an improved ability to maintain and debug the dataset. The volunteers that contributed said that they enjoyed the process.

Singh et al. (2002) [2] in their paper described the results they got by analyzing the original Open Mind Common Sense model (OMCS-1) and how they created the next generation of the system (OCMS-2). Firstly, to manually evaluate OMCS-1's quality and composition, they collected 3245 unique items from the dataset, which is just 1% of the 400,151 items. From these 3245 items, 236 (7.3%) items were automatically discarded because they required additional information such as images to be understood. The rest 3009 records were analyzed by 7 judges, who discarded 370 more items as they made no sense. The rest 2639 items were ranked on a criteria of 1 to 5 as such: sense (1= makes zero sense, 5= makes absolute sense), truth (1= false, 5= true), generality (1= fact, 5= general truth) and neutrality (1= biased, 5= not biased). Mean ratings for generality, truth, neutrality and sense were 3.26, 4.28, 4.43 and 4.55 respectively. The judges rated the sentences for age level and found out that 84% items were of high school level knowledge, which indicates that records in the database are well-known by most people. Then, the lessons learned while implementing OMCS-1 were used to design OMCS-2, in order to overcome the deficiencies of the former model. Some of the main factors learned are that templates are the most effective way of gathering data from users. Different volunteers liked to add different kinds of data, and they all wanted the engagement to be more

lively and wished they could access, clarify, repair and work further to arrange the entered knowledge. Therefore, in OCMS-2, the users were presented with options that they could choose from such as which type of knowledge they would like to enter. These entries were then spell-checked, POS tagged and disambiguated. In order to restrict the vocabulary used, synonym dictionaries such as WordNet were used, which suggested synonyms for uncommon words, which the users could accept or reject. In order to disambiguate the entries, sense tags were suggested, which could be modified by users. After the user corrected a few sense tags, the system was able to disambiguate the remaining words. In order to ensure that users entered the correct knowledge, OMCS-2 supported peer review, such that users could rate other users' entries and give them 'trust ratings'. The entries made by volunteers with higher ranks were provided with higher weights. In addition, the system checked if the entries are syntactically correct and if not, it raised an alarm and posted the doubtful entries for evaluation. OCMS-2 allowed evaluated records to be rectified and subjected them to more examination.

Havasi et al (2007) [4] have worked on ConceptNet 3 which focuses on the usefulness of the data in the OMCS project to natural language processing and artificial intelligence as a whole and it aims to make it modular in such a way that it can create conceptnets for various languages and synthesize them into the same database as ConceptNet, mainly English. In order to do that in this paper they brought some improvement factors such as higher-order predicates, polarity, and improved weight matrices. Open Mind Commons is the new updated version of Open Mind Common Sense data collector. It uses the data of OMCS and asks its user to give feedback so that it can get better data, another form is by asking users to fill in the blanks to gather data. For feedback if it found some concepts share the same multiple predicates but in a predicate it doesn't happen then it asks for feedback for that predicate so that it can get stronger concepts. For fill in the blanks, if some concepts don't know enough predicates then fill in the blank asks the user to fill in predicates. For initially creating predicates, it takes many texts and finds a pattern. If it finds a pattern, it is called "raw predicate" then it goes through normalization for creating proper predicate for ConceptNet. Data set consists of predicates and concepts and it got some scores to its predicates which showcases their reliability. One type of reliability score is predicate score. When multiple users use the same predicate multiple times for the same concepts, each time that predicate gets a point, sometimes evaluators increase the points of predicates this shows how accurate that predicate is. Another score for predicates in polarity, is used for detecting neglations, which is an additional pattern checking method which assigns the polarity value of predicates. In the analysis part the analysis has been done of ConceptNet3 predicates (IsA, PartOf, UsedFor, Random) with WordNet and Brandeis Semantic Ontology (BSO). In the analysis "Hit" means that concepts with predicates have that relation. IsA predicate got 45%(WordNet) and 42%(BSO), PartOf got 33%(wordNet) and 35%(BSO), UsedFor got 20%(BSO) and Random got 4%(WordNet) and 4%(BSO), In the end , it can be said that ConceptNet3 overlaps with renowned two resources where they are comparable with those resources.

Speer and Havasi (2012) [5] wanted to make the storage and querying of information truly distributable. The early ConceptNet procedure involved a lot of code and the query process was also time-consuming. Moreover, the projects on other websites continued to use their databases' out-of-sink versions. ConceptNet 5 is free from arbitrarily assigned IDs. The primary problem with ConceptNet3 is that the only way to access it is to sign from the same Django database. ConceptNet5 overcomes this problem by creating separate data from the interface. It provided multiple views on any particular word and how it can be structured around sentences. Computer application that has a brief understanding of natural language is supplied with a large set of contexts. The task is not just to predict lexical meanings but also to understand the connections between them using intuition. Data was collected from the website when visitors visited the sites. The websites asked the visitors targeted questions about statements they thought may be true. Similar information was also collected from the website when the visitors visited it in other languages e.g. Dutch, Portuguese, Japanese, and Chinese. The existing source of knowledge for ConceptNet5 is from the website. Statements were also gathered from several projects related to this topic in languages like Dutch and Portuguese. Translations of the multilingual data were also gathered from GlobalMind. Japanese statements were extracted from gwap.com and nadya.jp. Similarly, Chinese statements were also collected from PTT. Translations and synonyms were gathered from the English Wiktionary. WordNet 3.0 and Wikipedia articles represented in DBPedia have been used as the source of data collection. Relational statements are gathered from Wikipedia. Such enormous data collection resulted in a huge semantic graph. It portrays common human knowledge. As a result, a hypergraph is produced in which nodes are connected by edges. The nodes represent context and the edges show how the context and concepts are linked to each other. Furthermore, the edges also have weights assigned to them which can be negative or positive. ReVerb gathers the most important and valuable data from a dataset of front-paged Wikipedia entries. Negative edges were also removed in the process. MongoDB and Apache Solr was used as a source of indexing for ConceptNet5. People were asked to select between "Generally true", "Somewhat true", "I don't know", "Unhelpful or vague", "Generally false", and "This is garbled nonsense". They analyzed 81 responses from 1888 statements that were collected. The total statements were also reduced to 1193 since statements containing "Don't know" were removed. In Existing ConceptNet, WordNet, Verbosity, and both Wiktionary, English-only got the highest vote for "True". In conclusion, Wiktionary (translations), DBPedia, Reverb, and Global Mind Translations got the highest vote for "Don't Know".

In their paper, Trinh and Le (2019) [7] offered a staightforward approach for commonsense reasoning using unsupervised learning and neural network. This approach is used mainly for pronoun disambiguation problems that require commonsense knowledge to be made sense of. The model used language models to compute the probabilities of the possible outputs and then finalized the output with the greatest probability ratio. Two types of recurrent language models were used, in which one processed word inputs and the other processed character inputs, and both the models gave word outputs. For the word input model, the vocabulary size was 800K and embedding size was 1,024. For the character input model, the vocabulary size

was 256 and embedding size was 16. They found out that language models trained on large-scaled unlabeled data can hold a wide range of natural language and other types of knowledge, particularly common sense related data. They performed experiments on several different text corpora to investigate how the test accuracy is effected by the type of training data. Winograd Schema multiple choice questions and Pronoun Disambiguation Problems were then used to test the model. The results for Pronoun Disambiguation Problems portrayed that the ensemble of the five unsupervised models had an accuracy of 70.0%, which overpowered that time's best model's accuracy of 66.7% which used a supervised deep neural network and three knowledge bases. Full scoring, Full Normalised and Partial scoring were used to see which performance measure gives us the best result, and after comparing the results for 10 language models, it was found that for most of the language models evaluated, partial scoring performs better than every other method. While comparing the text corpora, it was found that STORIES had the highest accuracy for both types of inputs. Overall, the model achieved 63.7% accuracy over that time's state-of-the-art model that had 52.8% accuracy.

Talmor et al. (2019) [6] presented COMMONSENSEQA, a dataset concentrating on answering commonsense questions by collecting information from CONCEPTNET. It followed a process for creating questions for commonsense by using volunteers to ask questions that presented the relation among two concepts fetched from CONCEPTNET. For generating the data they first collected subgraphs from CONCEPTNET by filtering triplets(c1,r,c2) with general relations and by filtering where one of the concepts consists of more than four words and where the space from c1 to c2 is too low. To create multiple-choice answers for the same concept and relation 3 triplets are generated from the data. After these sets are created, the crowdworker creates a question from the sets where one is the correct answer and the other two are wrong answers. After that additional two distractors are inserted as an option where one is generated from other concepts from the same relation and the other is manually inserted by the crowdworker where the distractor is nowhere near the main answer. After that, the questions are verified by other crowdworkers which filters out many meaningless questions. To answer the question some context is added to it. After this dataset was ready which has multiple choice questions where there are 5 options and only one is the correct answer. In the dataset there is 44% of the first words are WHwords and 5% of the questions utilized first names to generate the story for context and in a hypothetical question "if" word is used for about 7%. This QA dataset has about 12,247 questions and seeks to asses commonsense knowledge capacity and it is seen that the best model obtains 55.9% accuracy.

Kejriwal and Shen (2020) [8] focused on finding the generalization in fine-tuned commonsense knowledge Models, for that result they judged the model on five benchmarks. aNLI (abductive Natural Language Inference) this benchmark checks the possible explanations for given observations. HellaSwag this benchmark works by giving multiple choice solutions for contexts and chooses the correct one. PIQA Physical Interaction QA this dataset gives prototypical and non-prototypical uses of a given object. Social IQA gives human-curated and machine generated solutions

for any situation. CycIC creates multiple answers on any question from the knowledge of its dataset. In this paper the data is taken from RoBERTa.The average score here is 80% accuracy. RoBERTa model is tested in every benches but in this test they are trained in one benchmark and tested in another benchmark to find the accuracy. For every benchmarks, the test on that benchmark is approximately 80% but the accuracy for aNLI is 0.819, 0.611, 0.702, 0.531, 0.442 for HellaSwag is 0.681, 0.835, 0.699, 0.515, 0.351 for PIQA is 0.68, 0.564, 0.756, 0.51, 0.371 for Social IQA is 0.688, 0.604, 0.687, 0.769, 0.508 and for Cyc is 0.628, 0.49, 0.628, 0.493, 0.811 respectfully on aNLI, HellaSwag, PIQA, Social IQA, Cyc. This number showcases that the accuracy of training and testing in those benchmarks is approximately 80% but when trained and tested in different models the accuracy is very poor, as the accuracy is poor the Performance Loss is high in these situations also. There is no pattern in the accuracy which shows that the benchmarks work biasedly on datasets. The performance drop is visible in different benchmarks for RoBERTa models, so these language models are not generalized because if they were generalized then the accuracy would have been closer.

In their paper, Shwartz et al. (2020) [9] proposed an unsupervised framework based on self-talk as a novel alternative to multiple-choice commonsense tasks. They wanted to show that even with knowledge bases and supervised learning, the models could not get the predictions right. The large model needs more coverage and precision. Even though the language models are quite efficient at predicting the semantic classification of an absent word, there is a possibility that the inaccurate examples were guessed right due to limited reasoning capabilities. External knowledge warrants discerning relevant and helpful facts that were integrated into language models. Language models also required to determine if an estimation is accurate in terms of factual information. They revealed that in the perspective of humans a large number of the clarifications even those that improved the prediction as being ineffective or inaccurate, proving that LM-based models frequently find accurate answers to scenarios that have the wrong answers. The ATOMIC dataset proved as the source of the knowledge. Crowdsourcing was used to get answers to the Physical Interaction Question Answering. WinoGrande was crowdsourced using a methodical process that produced a variation of instances that are trivial to people. The daily life scenarios were gathered from the Atomic responses. They were eventually categorized into several sentence classifications. They gathered a minimum of a hundred pairs of words from the context and question and the answer choices in Google N-grams. Supplementary resources include WordNet, retrieval or statistics from corpora, knowledge base embeddings, hand-crafted rules, and tools such as sentiment analyzers and knowledge-informed language models. They gathered relations within phrases from question and context. One language model was utilized as a knowledge source, while another language model served as the response scorer. Social IQa, Multiple Choice Temporal Commonsense, and Commonsense Question Answering were used to check if the language model predicted correctly or not. Every answer collected was allocated to a particular mix of questions and contexts. Therefore, the sentence with the most supportive clarification receives the highest point depending on the response option. In conclusion, based on the analysis, the overall performance of the unsupervised model is poor in comparison to

other supervised models. On the other hand, their performance is better for several tasks except for WinoGrande where they outperformed them a little bit.

In their paper, Ma et al. (2021) [12] posed the question of what models learn from commonsense reading datasets. Fine-tuning large language models is very common in solving some common sense problems. However, during their training stage, the models tend to overfit task-specific data and hence they forget the knowledge during pre-training. This issue has been studied and researched over generations by using more lightweight models. However, it is still difficult to accurately say which aspects and how much of a model should be improved for any particular task. Their paper focuses on how the models react to different datasets and the answers they produce in different scenarios. Their motivation was to test the performance of a few models on certain parameters and situations. To improve the models and keep them from being overfit, they will thereafter analyze their performance and compare it with that of other models. Three models of representative learning were presented by them: autoprompt for model prompting, regular fine-tuning, and model extension using prefixtuning. GPT-2 and BART are two example model classes to which the models were applied. ProtoQA and CommonGen, two standards for generative evaluation, were evaluated. Datasets from CommonGen and ProtoQA were gathered. In the instance of ProtoQA, the model was supposed to provide a ranked list of responses for a certain query. They calculated the answer matches using WordNet similarity and implemented ProtoQA's approved analysis measures. The information was shown as template-masked tokens, and a common sense knowledge graph was produced. CommonGen was tasked with using each of the three to five input notions it was instructed to build a scene description. Models were trained on two adaptation methods: a non-overlap subset of ProtoQA and a min-overlap split for CommonGen. Thirty of the ProtoQA questions with at least slightly accurate model responses were chosen for testing. To maintain consistency in the thinking process yet vary in the solutions, some of the questions were modified. Based on the task, BART models generated thirty new questions, which were then manually cross-checked with the original questions to prove their correctness. Prefix-tuning performed either comparably to fine-tuning or marginally worse at the adaption level. While Autoprompt outperforms the zero-shot baseline, it trails behind the adjusting process. Performance for task structuring and fine-tuning drops dramatically. Prefix-tuning results in a lower performance loss than fine-tuning. However, in this dataset, Autopromt performs comparatively better than both of them. As a result, their research indicates that prefix-tuning may more effectively generalize to novel ideas regardless of the task format, whereas prompting is the least susceptible of the three approaches to the quantity and quality of training data.

Xu et al. (2021) [10] proposed the descriptive knowledge for Commonsense question answering (DEKCOR) model in their paper. When the question and the choice is input in this model, it extracts the related concepts and the edge between the choice and the question from ConceptNet. If it is unable to find an edge, it calculates a relevance score for each knowledge triple which includes a object, relation and subject. The triple with the highest score is chosen. Wiktionary, which consists of

definitions of 999,614 concepts, is used to extract the definitions of these concepts by multiple criterias of text matching. The question, choice, selected knowledge triple and the definition are then put to the ALBERT language model to generate a result that represents how correct the answer is. There was a single model and an ensemble model. The latter model had seven single models which had dissilimar initialization random seeds and returned the choice that was chosen by majority of the embedded single models. CommonsenseQA and OpenBookQA are the two datasets on which this model was evaluated on and for CommonsenseQA, it outperformed the previous best model by 3.8% (ensemble model) and 1.2% (single model). For OpenBookQA, it performed better than all baseline models apart from two large-scale models which were based on T5. The research showed that the usage of concept descriptions from Wiktionary and knowledge triples from ConceptNet improved the accuracy of the model by 2.7% and 4.4% respectively for both the datasets, which proved that additional context information helps to fuse knowledge graphs into language models for commonsense question answering.

Yasunaga et al. (2021) [14] proposed a new model QA-GNN in their paper to overcome the issue of filling responses for questions. The information required to answer such questions is largely generated from large language models. The problem is that they acknowledge related information from large knowledge graphs. Language models are often applied separately to graph neural networks and the question-answering context and do not unify or update each model's representations. Their capacity to engage in structured reasoning may be hampered by this division. The QA-GNN model includes two components: joint reasoning, which connects the question-answering context and knowledge graph and uses graph neural networks to mutually update their representations, and relevance scoring, which uses language models for prediction of question answering. CommonsenseQA, OpenBookQA in the commonsense domain, and MedQA-USMLE in the biomedical domain were used as datasets for question-answering purposes. CommonsenseQA has 12102 questions and each question contains five multiple choices. OpenBookQA has 5957 questions and each question contains four multiple choices. The questions were based on scientific knowledge. MedQA-USMLE has 12723 questions and each question contains four multiple choices and the questions were related to biomedical sectors. They used ConceptNet as a source of knowledge graph creation for CommonsenseQA and OpenBookQA. The graph contained 2,487,810 edges and 799,273 nodes. In the example of MedQA-USMLE, they combined DrugBank and the Disease Database to make a self-constructed knowledge graph. There were 44,561 edges and 9,958 nodes in the knowledge graph. SapBERT was used for the initialization of node embeddings. Every question answering context was obtained, and the hop size of the subgraph was $k = 2$. The top 200 nodes are then retained once the subgraph is pruned. To get the question answering context, they concatenated a question and its corresponding answer. They established a node z for question-answering context which they connected to the topic entries. The primary distinction between QA-GNN and other models is their lack of joint updates with the question-answering context and relevance scoring. They used QA standards from the biomedical (MedQA-USMLE) and commonsense (CommonsenseQA, OpenBookQA) domains to assess their approach. QA-GNN demonstrates the ability to do organized and interpretable reasoning,

outperforming the current Language models and Language models with knowledge graph models. QAGNN performs significantly better in comparison to powerful fine-tuned language model baselines and Language Models with Language graphs by 4.7% and 2.3% respectively. In the comparison of language models with knowledge graphs, fine-tuned language models proved an improvement of +4.6%. While language models + knowledge graph models provided an improvement of +0.6% on queries that hold negation. They recommended that language models + knowledge graph system over RoBERTa; but QAGNN outperformed all of them with a significant +4.6% rise.

JointLK, which combines knowledge graphs and language models (LM) to solve commonsense question answering problems, was talked about by Sun et al. (2022) [13] in their paper. Commonsense knowledge extraction from language models is usually done by retrieval of subgraphs by string matching or considering semantic similarity. These extractions include relations between concepts and are modeled by graph neural networks (GNN) to form knowledge graphs. This process is problematic as the extracted knowledge subgraphs consist of noisy nodes and there are limited interactions between the knowledge graph representation and the language model. JointLK is used to overcome these problems. After JointLK receives a question and the retrieved subgraphs, it obtains two representations by using a GNN encoder and a LM encoder. It then creates bidirectional attention maps between the knowledge graph nodes and the question tokens to model the right path that outputs a real number that represents the correctness score of the answer. A dynamic graph pruning method is used to prune irrelevant nodes. Multiple layers of such models containing the GNN encoder, dynamic pruning module and joint reasoning module are stacked to support recursive pruning and multi-step interactions. In this paper, the performance of the model was judged by its performance on multiple choice questions, where the questions did not have relevant clues about the answers, compelling the model to rely on information from the knowledge graph. Two datasets over which the module was evaluated on are: CommonsenseQA and OpenBook QA, which are 5-way and 4-way multiple-choice question answering datasets respectively. During pre-processing, the max hop size was 3. The text encoders were configured with a largest input sequence length of 100 and used Cross-entropy loss and RAdam optimisers. The batch size was 128, dimension was 200, number of layers was 5 and early stopping was performed. The dropout rate was set to 0.2 for each layer. Separate learning rates were used for separate encoders. The results and comparison with similar models showed that JointLK functions the best within every LM+KG models and fine-tuned language models. The model's functionality upgrades by 5.74% on fine-tuned language models and 1.02% on the previously best LM+KG model (QA-GNN) over CommonsenseQA dataset. On the OpenbookQA dataset, its performance improves by 6.52% over fine-tuned AristoRoBERTa, and 2.15% over QA-GNN. The dynamic pruning mechanism and joint reasoning's effectiveness is reflected by the model's better performance over QA-GNN and the Multi-Hop Graph Relation Network (MHGRN), which previously performed better. Further analysis showed that increasing the number of layers improved the performance till there were 5 layers. After that, an increase led to a drop in performance. Retention ratio, which is a hyperparameter of the dynamic pruning module, showed

almost no pruning effect when set to a high value. K= 0.92 worked the best for the CommonsenseQA dataset.

According to Mallen et al. in 2022 [16], large language models still struggle with assignments that involve real world knowledge. They said that the factual knowledge that is widely spread throughout the internet is well-memorized by the large language models, but the information that is not so popular may not be well-known by these models and thus, retrieval from external sources might be needed. The dataset POPQA consists of 14 thousand questions related to factual knowledge that are not so popular. Even GPT-3 failed to answer the majority of these long-tailed questions. The authors randomly took triplets from Wikidata, consisting of 16 different relation types. The triplets were in the form (subject, relationship, object), and had varying degrees of popularity. The subject's popularity was based on its monthly Wikipedia page view. The triplets were then used to form questions using some templates. The relations between the subject and the object were also taken into suggestion as it was also suggested that pairs that correspond to some relations might be memorized faster by the large language models, based on that relation's popularity. Two datasets named POPQA and EntityQuestions were used in their research. Ten large language models were evaluated in zero-shot and few-shot prompting manners, and the answers were considered correct only if the actual answer exists as a substring of the answer generated by the large language models. For the retrieval-augmented approach, BM25 and Contriever retrieval systems were used to retrieve context from Wikipedia. These contexts were then concatenated with the existing questions to form the answers. As the large language models struggle with the subjects with low popularity, this approach helps the models significantly in such cases. A parametric augmentation method called GenRead was used to prompt the large language models to generate context. This did not improve the performance of smaller models, but showed slight improvement in the performance of larger models. Therefore, the authors found out that retrieving non-parametric memories is better for the smaller models as a small model supported by the retrievals from Contriever outperformed the vanilla GPT-3 model. Moreover, retrieval augmented language models outperform the language model's parametric memory for subjects with low popularity, while the parametric memory for the subjects with higher popularity are competitive. This is because results through retrieval are not always correct and can easily mislead the large language models. Also, smaller models retrieve more than larger models. Adaptive retrieval is a method suggested by the makers of POPQA and this method suggests that we should only retrieve knowledge when necessary. This means that retrieval must be done only for the questions whose subject's popularity is lower than a popularity threshold. This threshold is chosen to maximize the adaptive accuracy which is measured by comparing results by retrieval for less popular subjects and results based on the parametric knowledge of the large language models for the more popular subjects. This approach outperformed all other approaches and reduced latency and API costs.

Large language models are often biased towards Western culture and knowledge since the models are not trained with appropriate cultural differences for their respective languages. Hence, in their 2023 paper[19], Naous et al. created CAMeL, a framework of 628 prompts and 20368 entities classified into 8 types containing

varying Arab and Western cultures. It is a framework to determine the cultural bias of large language models towards Arab and Western cultures. The eight classifications of 20368 entities are person names, food dishes, beverages, clothing items, locations, authors, religious places of worship, and sports clubs. The entities were collected from Wikipedia manually and extracted from the Arabic subset of the Common Crawl by pattern-based entity extraction. The entities were classified into either Arab culture, Western culture, other foreign culture, not culture-specific, or non-entities. The Cohen's Kappa for the human annotation is 0.927 and also 15-20% of common crawl entities overlap with Wikipedia. The CAMeL prompts were split into culturally-contextualized prompts (CAMeL-Co) and culturally-agnostic prompts (CAMeL-Ag). In the case of CAMeL-Co, naturally context prompts were collected from Twitter/X and the tweets were between 8/1/2023 to 9/30/2023 so that large language models are not overtrained. The original context entities from the 250 CAMeL-Co prompts were replaced with a [MASK] token. Similarly for CAMeL-Ag, 378 prompts were collected by using generic patterns as search queries to avoid cultural reference. The tasks based on which the monolingual and multilingual large language models were evaluated were sentiment analysis, story generation, named entity recognition, and text infilling. For sentiment analysis, the annotators labeled the term either positive, negative, or neutral and the Cohen's Kappa is 0.954. For sentiment analysis, CAMeL-Co was used as testing data where the [MASK] token was replaced with 50 random Arab and Western entities. The HARD dataset was used for sentiment analysis. The result shows higher false negatives on sentences with Arab terms suggesting that the language models considered Arab terms with negative sentiments. For named entity recognition, the sentence with person names or locations and the ANERCorp dataset were used for evaluation. The result shows language models were more successful in tagging Western names or locations rather than arab ones. In story generation, the large language models were prompted to generate a story for a given Western or arab name. The adjectives from the story were collected by using the Farasa POS tagger. The frequency of adjectives was gathered and the Odds Ratio was computed. For Western stories, a larger Odds ratio was found rather than arab stories. Lastly, in text filling, they used cultural bias scores to determine the degree of Western bias. The score was around 40-60% in CAMeL-Co which is very high as CAMeL-Co was specially manufactured based on cultural influence. The results show that Arabic monolingual models show some level of Western bias whereas multilingual models exhibit a higher level of Western bias.

# Chapter 3

# Workplan

In Pre-Thesis 1, We formed a team and applied to our supervisor for supervision. After confirmation, we did some study and research on Natural Language Processing and started the work for Pre-Thesis 1. We selected this topic and the title was narrowed down. After that, we were given some papers regarding our topic, and we read those and noted the summaries. We were in touch with our supervisor weekly and reported our paper-reading activity. We wrote the abstract and submitted it.

In Pre-Thesis 2, we web scraped some data to collect terms related to the Bengali language. This data was then used to crowdsource information in order to create a dataset. After analyzing the crowdsourced data, we came to the conclusion that the data we collected was of poor quality which encouraged us to take an alternative approach.

In the Final Thesis, we developed a corpus to evaluate the performance of large language models by masked prediction and question answering. For question answering, we checked the accuracy of two large language models, based on appropriate popularity benchmarks. In masked prediction, the knowledge of the large language models were analyzed across different categories that we formed within our dataset. In the report, the results were analyzed and finally completed the final paper of our thesis, and prepared for our thesis defense.

The figure 3.1 represents the flowchart of our thesis.

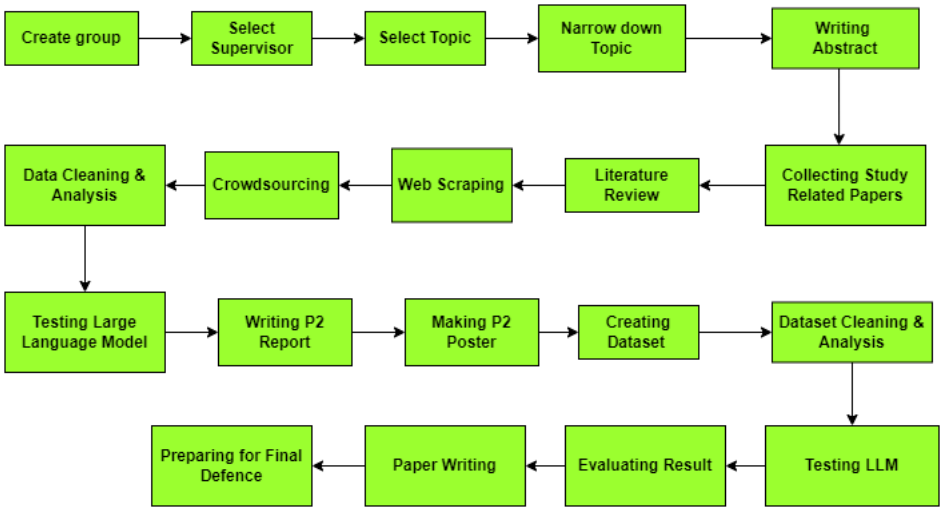


Figure 3.1: Thesis Flowchart

# Chapter 4

# Dataset

## 4.1 Crowdsourcing

Our initial approach was to measure how much knowledge the large language models are composed of by prompting them with pairs of terms and their classifications along with some relationship types. We would ask the large language models to choose the most appropriate type for each term and the relationship between them. The term types we considered were 'Event', 'Location', 'Temporal', and 'Miscellaneous'. The classifications for relationships were 'Functional', 'Spatial', 'Taxonomic', 'Conceptual', and 'Not Applicable'. Firstly, we needed to have such a dataset and we decided to create the dataset through crowdsourcing.

### 4.1.1 Web Scraping

In order to get all the necessary terms needed for crowdsourcing, we scraped Wikipedia initially. For the basic terms, we scraped the Wikipedia page which had a list of every information related to **"বাংলাদেশ"** and extracted every title of other Wikipedia pages that are linked to it and collected their links. As we are working with the Bengali Language, we considered **"বাংলাদেশ"** page to be a source of some other Bengali terms which are connected to Bengali culture. In the same way, we scraped every Wikipedia page whose links we collected from the **"বাংলাদেশ"** Wikipedia page and collected links that are present in those pages and extracted their respective titles as terms. For instance, from **"বাংলাদেশ"** Wikipedia page, we got **"সুন্দরবন"** and **"মুঘল_সম্রাজ্য"** as titles with links and we collected them and again for these titles we again scraped these pages and collected every link consisting title as our term. In our web scraping process, we avoided titles that do not have any article in them. After collecting the terms, the terms which contained more than one word had underscores (_) in between the words so we replaced the underscores with spaces. For instance, for the term **"মুঘল_সম্রাজ্য"** had underscore in between them so we replaced the underscore with space like **"মুঘল সম্রাজ্য"**. We made a network graph

where each of the terms were nodes and were connected to the nodes that represent the pages they were collected from. We generated that graph in the hope that it could be used as a knowledge graph as many Bengali terms were present there.

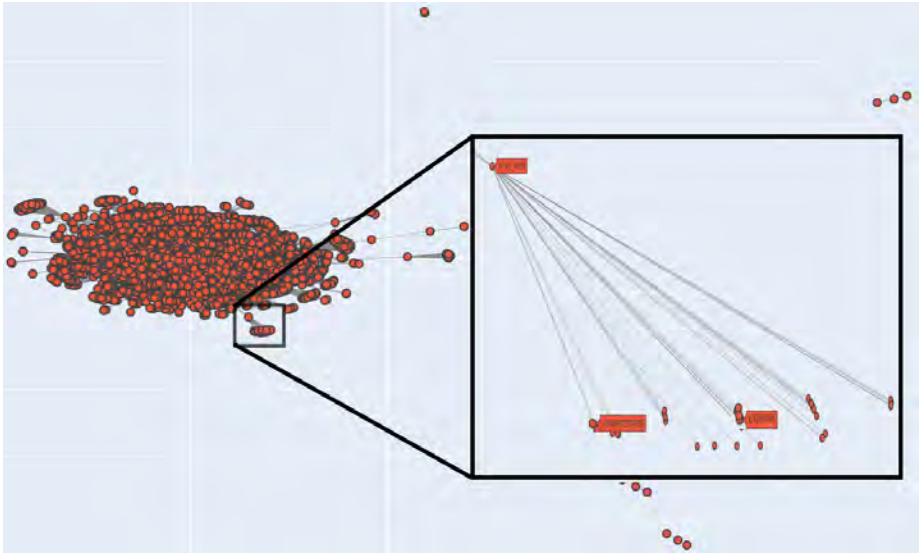

Figure 4.1: Network Graph

The graph had 6,712 nodes and 12,076 edges. The figure 4.1 shows the whole graph and shows the small portion that has been zoomed on where the node "বাংলা ব্যান্ড" is connected to nodes like "ওয়ারফেজ" and "মেঘদল".

### 4.1.2 Crowdsourcing through Website

To collect important terms we did crowdsourcing through a website. To begin with the crowdsourcing, we needed to provide the annotators with some pairs of terms. In order to create these pairs, we scraped these 6,702 links and found out the most important terms from these articles using TF-IDF and then created clusters. Then, we manually went through these clusters to find out terms that had no direct connection, that is, we do not immediately think of one term when we hear of another term. We ensured that the terms have a multi-hop relationship because we were not looking for factual information. After handpicking the pairs of terms with multi-hop relationships, we provided them to our annotators. In the annotation platform, the first two questions were to select the type of the terms from the given types. The third question was to select the relation between two terms from the given options. The next two questions were to provide terms that are related, similar, or opposite of the given two terms. The final question was to construct a sentence containing both terms. Each annotator was asked to do all of these tasks for three pairs of terms. Initially, we wanted to test our experiment on a smaller scale. The dataset that we collected through crowdsourcing consisted of 285 rows and 8 columns. After cleaning the data set, we have 256 rows and 8 columns. The dataset contains some categorical data and some text data.

| Term1 | Term2 | Type Term1 | Type Term2 | Relation | Term1 words | Term2 words | Sentence |
|---|---|---|---|---|---|---|---|
| বাউল | লুঙ্গি | a)event | a)event | a)functional | একতারা,দোতারা | প্যান্ট,পায়জামা | বাউল শিল্পীরা সবসময়ই লুঙ্গি ব্যবহার করে থাকেন। |
| জামদানি | রেশম | a)event | a)event | a)functional | কাপড়,আরাম | দোকান,সুন্দর | এই দোকানে রেশম জামদানি পাওয়া যায়। |

Table 4.1: Sample of Crowdsourced Dataset

Table 4.1 shows a part of our crowdsourced dataset.

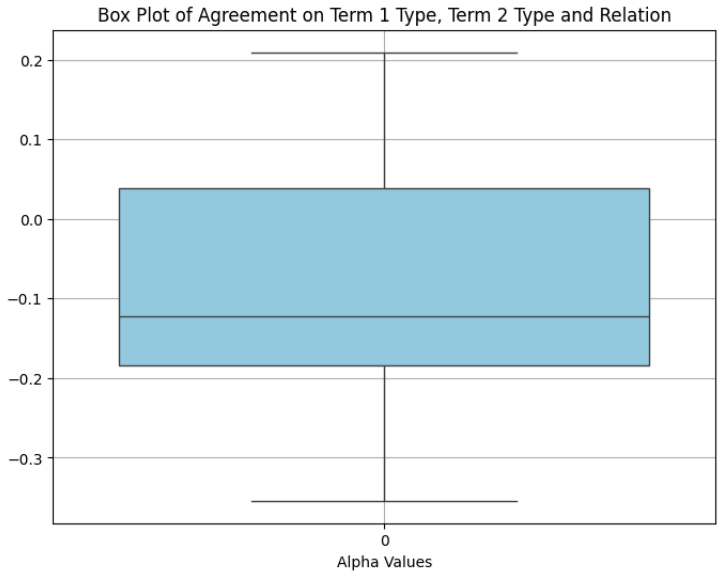


Figure 4.2: Inter Annotator Agreement

The data quality was not that good as there was ambiguity. For example, for the pair containing the terms '**বরপক্ষ**' and '**যৌতুক**', some annotators chose the relationship between them to be functional and some of them selected it to be conceptual. We found a similar situation for the pair that contains the terms '**জামদানি**' and '**রেশম**', where annotators chose both taxonomic and functional relations for the pair. Also, we measured the Inter Annotator Agreement of the data using Krippendorff's alpha, and as we can see in figure 4.2 the alpha values showed almost no agreement. As we have multiple relations types for the same pair of terms, we would not be able to check the accuracy of large language models without ambiguity. For a smaller scale, we found huge ambiguity which would have been worse if the experiment had been conducted on higher scale. Therefore, we took the alternate route to create a new dataset without the help of crowdsourcing.

## 4.2 Cultural Difference and Commonsense Knowledge Base

ConceptNet is a knowledge graph that has words and phrases connected through edges representing the relationship between them. As the Bengali language is mostly practiced in the South Asian part of the world, we checked the South Asian knowledge of ConceptNet. We extracted sentences related to cricket from ConceptNET as we wanted to identify the cultural differences. This approach is inspired by the paper "Can Common Sense uncover cultural differences in computer applications?" (Anacleto et al., 2006) [3]. Firstly, in order to check whether the large language models consist of enough commonsense to answer our queries, we wrote 75 Bengali sentences on the topic 'Cricket'. Among these, the sentences equally belonged to the categories of Taxonomic, Spatial, Causality, Functional, and Planning as mentioned in the Open Mind Common Sense (OMCS) project (Singh et al., 2002) [1]. Then, we found out the subject, object, and predicate of these sentences. These sentences were then translated into English before starting the test. After getting the sentences we first filtered out the sentences which had more than 50% cosine similarity between ConceptNet sentences and the 75 sentences. We used a human-in-loop verification pipeline again and there was only 7.95% similarity which represents a huge gap in cultural knowledge.

## 4.3 Dataset Construction

As the crowdsourcing approach failed, we took an alternate route to create a dataset that can be used for both masked prediction and question answering using large language models to evaluate knowledge of these models. To create this dataset we took the 6,702 terms we collected during initial web scraping and scraped context for each term from Wikipedia, similar to CAMeL's approach [19]. In CAMeL, the contexts were extracted from Twitter, but since in the Bengali language, there is no robust toolkit to mine the data so we had to adapt to scraping Wikipedia for context. These contexts mostly consist of the definitions of the terms, and are factual knowledge. Moreover, in a way similar to POPQA [16], we wanted to judge whether the performance of the large language models for each term depends on its popularity. Therefore, we created two popularity metrics. The first popularity metric is the monthly Wikipedia page view, where we averaged each Wikipedia page's monthly view of the past year. The other metric was the Term Frequency Inverse Document Frequency (TF-IDF) value of each term. We scraped the entire article of the Wikipedia page related to that particular term and appended the document to a list. This process was followed for every term in our dataset. Hence, the list consisted of the document for every term in our dataset. To calculate Term Frequency (TF), we measured the frequency of terms in that particular document and divided it by the number of words present in that document. To calculate Inverse Document Frequency (IDF), we took the logarithmic function of the total number of documents divided by the number of documents containing that specific

term. Then we multiplied the TF and IDF of that particular term to get its TF-IDF value. Both of these metrics are normalized by the Min-Max scaler to convert into a normalized form in our dataset to remove any sort of bias. The category of each term was determined by clustering using a community detection algorithm.

For question answering, we provided these terms and their corresponding contexts to IndicBART and C4AI to create questions whose answers are the terms themselves. Upon inspection, we observed that the quality of these questions varied for respective terms. From the table 4.2, we can observe that for the term 'রুটি', the question generated by the IndicBART is 'রুটি কিসের জন্য ব্যবহৃত হয়?'. In the case of IndicBART, the term রুটি was present in the question. This means that the large language models will already be supplied with the answers which will confuse them resulting in fluctuations in accuracy since we cannot determine whether the large language model answered the question with their own knowledge. On the other hand, for the term 'রুটি', the question generated by C4AI is 'কোন খাবারটি আঁটা বা ময়দা এবং পানির মিশ্রণ থেকে তৈরি করা হয় যা বাফাডে খাওয়া যায় এবং সহজেই সংরক্ষণ করা যায়?'. Hence, the quality of the question generated by C4AI was better than IndicBART since the term was not present in the question generated by C4AI. The table 4.2 shows the comparison between questions generated by C4AI and IndicBART for respective.

| Term | Question by IndicBART | Question by C4AI |
|---|---|---|
| রুটি | রুটি কিসের জন্য ব্যবহৃত হয়? | কোন খাবারটি আঁটা বা ময়দা এবং পানির মিশ্রণ থেকে তৈরি করা হয় যা বাফাডে খাওয়া যায় এবং সহজেই সংরক্ষণ করা যায়? |
| লঙ্কা | কোন রাজ্যগুলি সাধারণত রামায়ণে বর্ণিত রাজ্যকে নির্দেশ করে? | রামায়ণে বর্ণিত রাজ্যের নাম কী? |
| পাপ | পাপের অর্থ কী? | যে কোন ধরনের মন্দ কর্ম কি হলো? |

Table 4.2: Comparison of questions given by IndicBART and C4AI

Since the quality of the question generated by C4AI was better than IndicBERT, so in our dataset we included the question generated by C4AI only for respective terms. Our dataset includes terms, questions, contexts, normalized page view popularity metric, normalized TF-IDF popularity metric, and category in which that particular term belongs. The dataset we created for question answering consists of 6,533 rows and 6 columns. Table 4.3 is a sample of our dataset for question answering.

| Term | Question | Context | Page View | TF-IDF | Category |
|---|---|---|---|---|---|
| হিন্দুধর্ম | ভারত এবং নেপালের প্রাচীন ধর্ম কোনটি? | হিন্দুধর্ম ভারতীয় উপমহাদেশীয় ধর্ম বা জীবনধারা | 0.133 | 0.017 | Religion |
| নামাজ | ইসলামি ধর্মে দৈনিক নিয়মিত ইবাদত কি? | সুন্নি ইসলামি ধর্মতত্ত্বসমূহ নামাজ বা নামায বা সালাত বা সালাহ ইসলাম ধর্মের একটি দৈনিক নিয়মিত ইবাদত | 0.095 | 0.068 | Religion |
| চাঁদ | সৌর জগতের পঞ্চম বৃহত্তম উপগ্রহ কোনটি? | চাঁদ পৃথিবীর একমাত্র প্রাকৃতিক উপগ্রহ এবং সৌর জগতের পঞ্চম বৃহত্তম উপগ্রহ | 0.090 | 0.057 | Miscellaneous |

Table 4.3: Sample of Dataset

For masked prediction, we used the same dataset and masked the term from their respective context. The masked context was provided to the large language models for mask filling. For terms consisting of more than one word, each word was masked separately creating multiple masked contexts. This approach causes the number of rows in our dataset for masked prediction to increase from 6,533 to 8,825. Table 4.4 is a representation of our dataset with the terms, context, and masked context.

| Term | Context | Masked Context |
|---|---|---|
| হিন্দুধর্ম | হিন্দুধর্ম ভারতীয় উপমহাদেশীয় ধর্ম বা জীবনধারা | [MASK] ভারতীয় উপমহাদেশীয় ধর্ম বা জীবনধারা |
| নামাজ | সুন্নি ইসলামি ধর্মতত্ত্বসমূহ নামাজ বা নামায বা সালাত বা সালাহ ইসলাম ধর্মের একটি দৈনিক নিয়মিত ইবাদত | সুন্নি ইসলামি ধর্মতত্ত্বসমূহ [MASK] বা নামায বা সালাত বা সালাহ ইসলাম ধর্মের একটি দৈনিক নিয়মিত ইবাদত |
| চাঁদ | চাঁদ পৃথিবীর একমাত্র প্রাকৃতিক উপগ্রহ এবং সৌর জগতের পঞ্চম বৃহত্তম উপগ্রহ | [MASK] পৃথিবীর একমাত্র প্রাকৃতিক উপগ্রহ এবং সৌর জগতের পঞ্চম বৃহত্তম উপগ্রহ |

Table 4.4: Sample of created Masked Contexts from the Dataset

## 4.4 Dataset Analysis

The terms in our dataset were categorized across 16 classifications. These classifications were determined with the help of the community detection algorithm which is an unsupervised algorithm to detect clusters. The classification includes the following categories: **location, miscellaneous, people, religion, entertainment, organization, food, politics, language, temporal, historical, event, material, health, sports,** and **economics**. These categories consist of both factual and Bengali culture-related classifications, which will help us analyze how the large language models perform in each of these fields and where improvement is necessary.

A brief description of each of our 16 categories along with the examples of terms present in them is given below:

- **Location:** This category consists of names of places and countries. Some examples of words present in this category are 'ত্রিপুরা',' শ্রীলঙ্কা', and 'সুন্দরবন'.
- **Miscellaneous:** The terms that do not fall in any of the other categories are considered miscellaneous. Some examples of such terms are 'পেশা', 'গণমাধ্যম' and 'ফিডব্যাক'.
- **People:** It consists of the names of famous people, mostly politicians, writers, singers, and many more, most of whom are from our subcontinent. Some examples of such people are 'শরৎচন্দ্র চট্টোপাধ্যায়', 'রশিদ উদ্দিন' and 'সম্রাট জাহাঙ্গীর'.
- **Religion:** This category consists of the names of things related to religion and religious terms. Some examples of terms belonging to this category are 'শিরক', 'পণ্ডিত' and 'দেবী'.

- **Entertainment:** The names of movies, music composer duos, etc. For example, 'সাজিদ-ওয়াজিদ', 'অবাক পৃথিবী' and 'ঘুড্ডি' are examples of terms present in this category.
- **Organization:** It consists of the names of many different types of organizations. 'নাসা', 'দেশ টিভি' and 'গুগল' are examples of terms belonging to this category.
- **Food:** The names of many different types of food and edible ingredients make up this category. Some examples are 'বেগুনি', 'আম' and 'চাল'.
- **Politics:** Names of politicians, political parties, and monuments of the Indian subcontinent are included in this category. 'মুহাম্মদ আলী জিন্নাহ', 'গান্ধী' and 'বাংলাদেশ আওয়ামী লীগ'are some names present in this category.
- **Language:** This category includes names of languages. Some names present in this category are 'মারাঠি', 'ফারসি ভাষা' and 'ইংরেজি ভাষা'.
- **Temporal:** Names of months, seasons, and calendars are present in this category. Some examples of terms from this category are 'হেমন্ত', 'সেপ্টেম্বর' and 'হিন্দু পঞ্জিকা'.
- **Historical:** Terms related to the history of the Indian subcontinent make up this category. 'বিজয় সেন', 'পশ্চিম চালুক্য সাম্রাজ্য' and 'পল্লব রাজবংশ' are some examples of terms in this category.
- **Event:** This category consists of names of different types of events that happened in our region including military operations, wars, etc. Some terms from this category are 'চীন-ভারত যুদ্ধ', 'অপারেশন জ্যাকপট' and 'বাংলাদেশের আন্তর্জাতিক স্বীকৃতি'.
- **Material:** Names of things we use in our day-to-day lives make up this category. 'কাগজ', 'চাদর' and 'চেয়ার' are some terms from this category.
- **Health:** It includes names of diseases, organs, vitamins, etc. 'ডেঙ্গু', 'চোখ' and 'ক্যান্সার' are examples of terms from this category.
- **Sports:** Names of players, games and tournaments make up this category. Some examples of terms from this category are 'ফিফা', 'ব্যাডমিন্টন' and 'সাকিব আল হাসান'.
- **Economics:** This category includes terms which are business related, are related to economics or directly influence the economy. Some examples of of terms from this category are 'আমদানী', 'বিশ্ব ব্যাংক' and 'টাকা'.

Figure 4.3 shows the percentage of terms in our dataset belonging to each of these 16 categories.

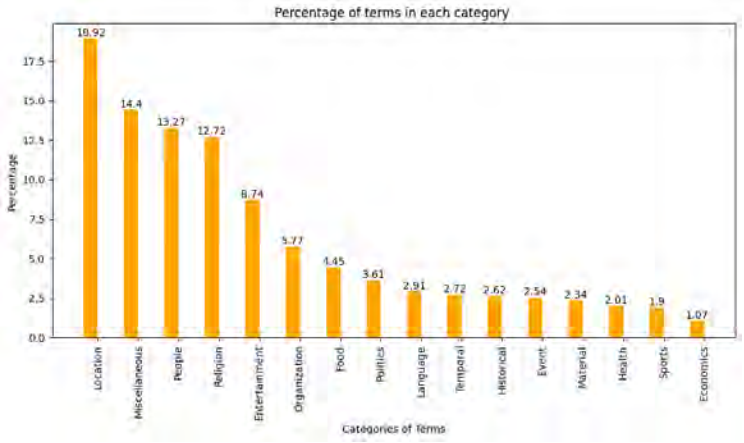


Figure 4.3: Percentage of each category in the dataset

In order to have a visual representation of the 16 categories and inspect how they are related in terms of their meaning and semantics, we used t-SNE. We used Bengali GloVe Vector Embeddings to vectorize the Bengali words. The distribution of the categories are observed in the figure 4.4.

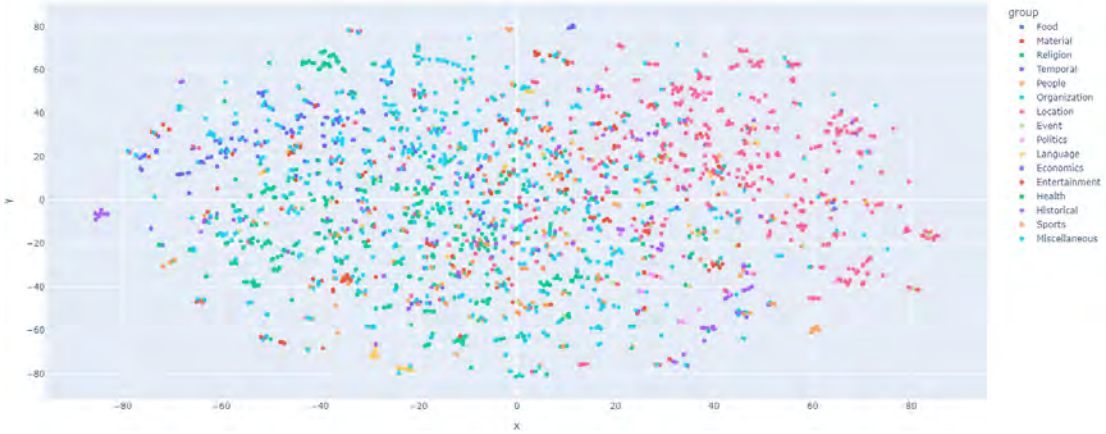


Figure 4.4: Visual representation of the terms using t-SNE representation

In the figure 4.4, each point represents a term in a color-coded manner to demonstrate the distribution of terms belonging to each category in our dataset. If a point is seen more in the t-SNE representation, it means that its presence in the dataset is prevalent and vice versa. The points close to each other in the visualization are close in meaning. In the representation some of the purple dots in the left represent the months of the Bengali year including 'পৌষ', 'আষাঢ়', and 'শ্রাবণ' . All of these are close to each other and they lie in the same cluster named 'Temporal'. This shows that the clusters we created are meaningful.

# Chapter 5

# Models

## 5.1 BERT (Bidirectional Encoder Representations from Transformers)

BERT is a transformer based model. The input provided to BERT is converted to embeddings. The multi-head self-attention mechanism concentrates on the context so it takes the relationship of the word with respect to the same sentences. Since BERT is a bidirectional model, it can read sentences from both directions which improves the performance of the models. In the multi-head self-attention phase, multiple query, key, and value are introduced for each head. In this way, the multi-head self-attention extracts the context and provides tokens to Feed Forward. The feed forwards add non-linearity and complexity to the tokens. The outputs are passed down to the next encoder and eventually passed to the next layers.

The final linear layer and softmax layer convert the vectors into a word. The final linear layer unifies all the attention heads. The softmax layer turns these vectors into probabilities. Using the argmax function the cell with the highest probability is chosen and the word associated with the probability is assigned. BERT is pre trained on Google's BooksCorpus and Wikipedia, and pretrained on unsupervised natural language processing tasks like masked language modeling and next sentence prediction (Muller, 2022) [17]. The figure 5.1 below shows the architecture of BERT.

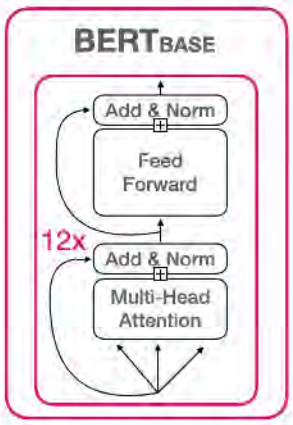


Figure 5.1: Architecture of BERT[17]

### 5.1.1 SahajBERT

SahajBERT is pre-trained on data from the Bengali portion of Wikipedia and OSCAR and the data was carefully curated (Jahan et al. 2023)[18]. It is based on the BERT models and focuses on the low-resource Bengali and Indian languages. This model consists of a tokenizer which is specialized for the Bengali language. The model is efficient for natural language processing tasks like masked language modeling and sentence order prediction (Jahan et al., 2023)[18]. We used SahajBERT for masked token prediction.

### 5.1.2 IndicBART

IndicBART is trained on IndicNLGSuit and fine-tuned from IndicBART and it is based on the ALBERT model (Dabre et al., 2021)[11]. The model is pre-trained using twelve Indian languages including Bengali, English, Hindi, Gujarati, and many more (Dabre et al., 2021)[11]. Therefore, the model is effective in Bengali question generation. So we used IndicBART to generate questions for each term in our dataset based on its context so that the questions were constructed in such a way that the answer to the question is the term.

### 5.1.3 BanglaBERT

BanglaBERT is another pre-trained transformer model for the Bengali language, and this was pre-trained using Bengali Wikipedia, news, government articles, websites, blog sites, newspapers and internet crawl (Kowsher et al. 2022) [15]. The dataset was very large, but the varying information in the internet meant that there was more noise in this dataset, as it was not properly curated and unannotated. This model was used for testing the existence of Bengali commonsense knowledge in large language models, and later for masked prediction using the dataset we constructed.

## 5.2 Open-source Large Language Model

### 5.2.1 C4AI

C4AI is an open source large language model making it open for everyone's use and modification. The size of C4AI is 35 billion parameters which determines how it processes and generates text. It has been trained with conversational tool capabilities hence which takes texts in the form of input and generates responses[21]. C4AI has been used to improve the question of the dataset since the question generated by IndicBART lacks in quality. Therefore, similarly, C4AI was prompted to give questions for each term based on context and the question should be manufactured in such a way that the answer to the question is that respective term.

### 5.2.2 Llama

Llama is an open-source large language model optimized for dialogue use cases. It is an auto-regressive language model that utilizes an optimized transformer architecture. It is aligned with human preferences since it is a tuned version of supervised fine-tuning and reinforcement learning with human feedback. Llama 3 has been pretrained on custom training libraries, Meta's Research SuperCluster, and production clusters [20]. The size of the Llama 3 version we used is 8 billion parameters, where it takes text as input and generates responses. We used Llama 3 for different natural language processing tasks like question answering and masked prediction. The architecture of the Llama model is shown in the figure 5.2.

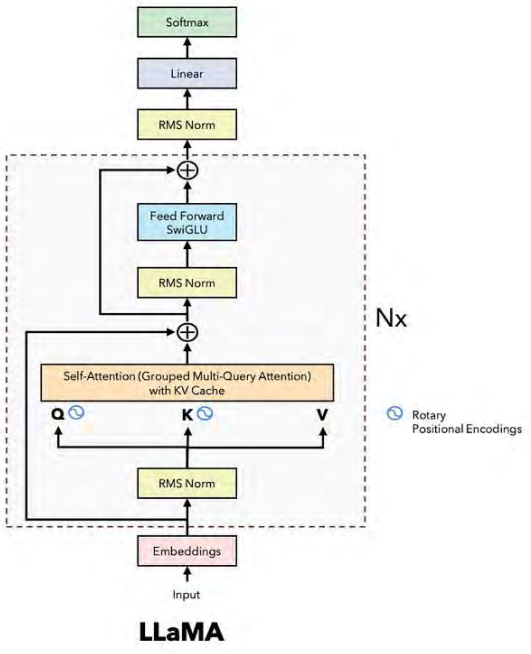


Figure 5.2: Architecture of LLAMA [23]

## 5.3 Closed-source Large Language Model

### 5.3.1 Gemini

Gemini uses a neural network architecture which is transformer model based. The model was designed for tasks with varying capabilities and use cases. It was pre-trained on data from several publicly available sources. Gemini's capability of performing many different tasks enabled us to use it for both masked prediction and question answering. The version of the model we used is Gemini 1.5 Flash which is designed to work more efficiently and is a lot more lightweight than other versions. It achieves almost a perfect recall for long-context retrieval tasks across modalities. Also, this model improves the state-of-the-art in long-document Question Answering, long-video Question Answering and long-context Automatic Speech Recognition (Reid et al., 2024)[22]. We accessed it using their API, as Gemini is a closed-source model. Since we used a free API to access the model, the rate limit is 1500 requests per day alongside 15 requests per minute and one million tokens per minute.

# Chapter 6

# Results and Analysis

## 6.1 Question Answering

In the question answering, we prompted the large language models Llama and GEMINI twice. Firstly, only the question was provided. Then, both the context and question were supplied to the models. To classify the popularities, we divided the dataset into 5 chunks. The first chunk contained the rows with the top 1000 popularity, the second chunk had the next top 1000 rows, and so on. The answers given by the large language models were considered accurate only if the term which is the actual answer is present as a substring of the generated answer. Since for the respective terms, the questions were generated based on context so we only accepted the term as the answer for respective terms. The accuracy given by the large language model is strictly matched with our answer since we want to determine the performance based on that particular term. The question answering was analyzed based on normalized TF-IDF popularity metrics and normalized Wikipedia page views popularity metrics. The description of the metrics is given in section 4.3.

### 6.1.1 Gemini

For Gemini, the overall accuracy for the result where only questions were supplied is 26.68% and the result where the context was included is 61.62%.

As shown in figure 6.1, after sorting the data in terms of the normalized tf-idf popularity metric, we found that for the prompting we did without context, the range of popularity 1-0.096 had 27.3% accuracy; the range 0.096-0.062 had an accuracy of 33.4%; the range of popularity 0.062-0.039 had 28.79% accuracy; the range 0.039-0.02 had 26.2% accuracy and finally the range 0.02-0 had an accuracy of 19.46%. Here we can observe that for prompting without context the results are close in every range except 0.02-0. As they have the lowest value of popularity it is expected for Gemini to perform badly for those terms. For the second time we prompted,

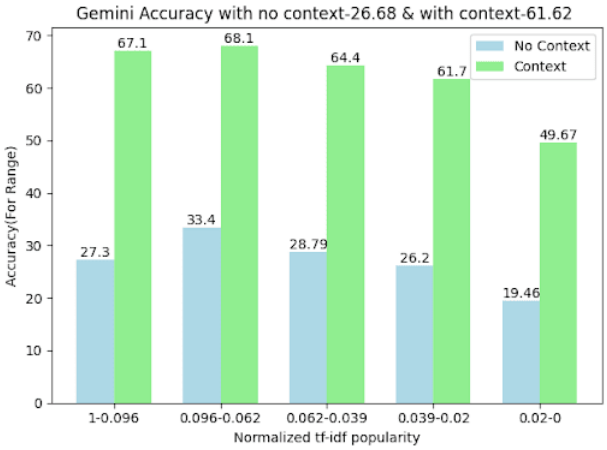


Figure 6.1: Accuracy of Gemini for Normalized TF-IDF Popularity

which included the context, the range of popularity 1-0.096 had 67.1% accuracy; the range 0.096-0.062 had an accuracy of 68.1%; the range of popularity 0.062-0.039 had 64.4% accuracy; the range 0.039-0.02 had 61.7% accuracy and finally, the range 0.02-0 had an accuracy of 49.67%. In the case of the popularity range between 1-0.096, Gemini underperformed in comparison with the popularity range of 0.096-0.062 and 0.062-0.039 in regards to without context. The range of 0.096-0.062 has the highest percentage accuracy in both with and without context though they are not the most popular by tf-idf popularity. In this scenario, we can say that there is a simple pattern that by the descent of popularity the quality of answers with and without context has also decreased except for the range of 0.096-0.062 but the accuracy with context is almost twice that without context.

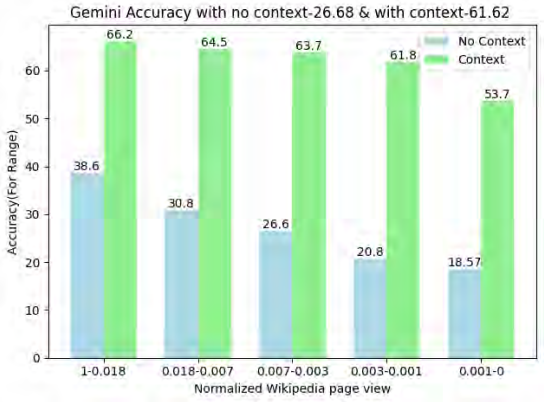


Figure 6.2: Accuracy of Gemini for Normalized Wikipedia Popularity

Similarly as shown in figure 6.2, after sorting the data in terms of the normalized Wikipedia monthly page view popularity metric, we observed that for prompting Gemini to answer the questions without context, the range of popularity 1-0.018 had 38.6% accuracy; the range 0.018-0.007 had an accuracy of 30.8%; the range of popularity 0.007-0.003 had 26.6% accuracy; the range 0.003-0.001 had 20.8% accuracy and finally, the range 0.001-0 had an accuracy of 18.57%. In this scenario, we can observe that with the descent in popularity, the percentage has also decreased which is expected as there are fewer views on Wikipedia then that term is less popular so the large language model may not have sufficient knowledge about that

term. Again we prompted, which included the context with the question, the range of popularity 1-0.018 had 66.2% accuracy; the range 0.018-0.007 had an accuracy of 64.5%; the range of popularity 0.007-0.003 had 63.7% accuracy; the range 0.003-0.001 had 61.8% accuracy and finally, the range 0.001-0 had an accuracy of 53.7%. The percentage is quite the same except in the range of 0.001-0, where the percentage has dropped a big amount when compared to other ranges, as the terms of this range may not be known by Gemini. Similar to the answers we got without providing the context, the accuracy decreases along with the decrease in popularity when context is provided.

### 6.1.2 Llama

For Llama, the overall accuracy for the result where only questions were provided to the model is 26.01% and 57.62% for without and with context respectively.

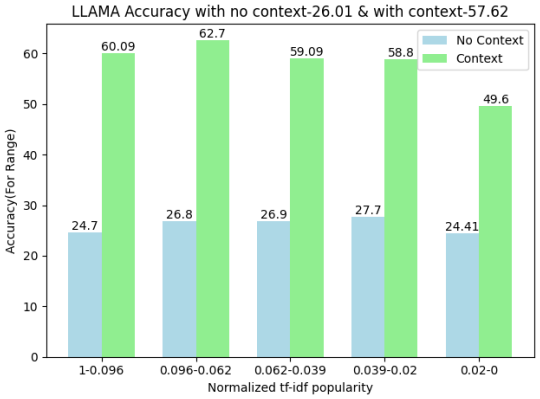


Figure 6.3: Accuracy of Llama for Normalized TF-IDF Popularity

From the above figure 6.3, after the data was sorted in terms of the normalized TF-IDF popularity metric, we observed the results for prompting Llama without context, the range of popularity 1-0.096 had 24.7% accuracy; the range 0.096-0.062 had an accuracy of 26.8%; the range of popularity 0.062-0.039 had 26.9% accuracy; the range 0.039-0.02 had 27.7% accuracy and finally the range 0.02-0 had an accuracy of 24.41%. Here we can observe that for prompting without context the accuracy has increased with the descent of tf-idf popularity except for the last range of 0.002-0. In other words, rather than the highest popularity or lowest popularity the in-between popularities gave better accuracy in answering without context. The popularity range 0.039-0.002 gave better results when compared to other ranges in terms of without context. For the second time we prompted, which included the context, the range of popularity 1-0.096 had 60.09% accuracy; the range 0.096-0.062 had an accuracy of 62.7%; the range of popularity 0.062-0.039 had 59.09% accuracy; the range 0.039-0.02 had 58.8% accuracy and finally, the range 0.02-0 had an accuracy of 49.6% Here the highest accuracy is in the range 0.096-0.062. The accuracies varied a lot by the range distribution resulting in no pattern.

Similarly, from figure 6.4, when the data is sorted in terms of the normalized Wikipedia monthly page view popularity metric, we observed that for prompting

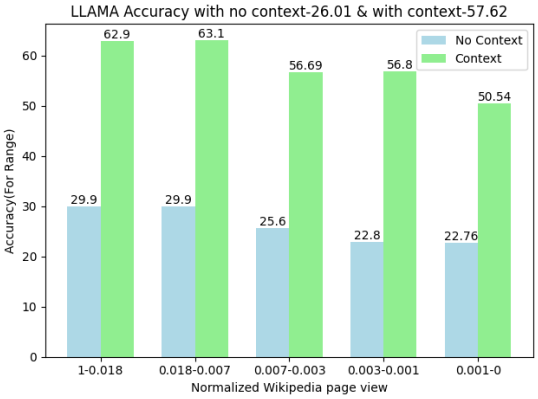


Figure 6.4: Accuracy of Llama for Normalized Wikipedia Popularity

Llama to answer the questions without context, the range of popularity 1-0.018 had 29.9% accuracy; the range 0.018-0.007 also had an accuracy of 29.9%; the range of popularity 0.007-0.003 had 25.6% accuracy; the range 0.003-0.001 had 22.8% accuracy and finally, the range 0.001-0 is also close 0.003-0.001 with an accuracy of 22.76%. From the above scenario, we can observe that with the descent in popularity, the percentage has also decreased which is expected as there are fewer views on Wikipedia so that term is less popular so the large language model may not have sufficient knowledge about that term. Again we prompted, which included the context with the question, the range of popularity 1-0.018 had 62.9% accuracy; the range 0.018-0.007 had an accuracy of 63.1%; the range of popularity 0.007-0.003 had 56.69% accuracy; the range 0.003-0.001 had 56.8% accuracy and finally, the range 0.001-0 had an accuracy of 50.54%. The popularity range 0.018-0.007 provided better results in comparison with other ranges despite it being the second most popular range. However, upon analysis of the outcome of Llama concerning context, we could not arrive at a fixed pattern.

### 6.1.3 Question Answering Analysis

For the TF-IDF popularity metric, it is evident that the range 0.096-0.062 has the highest accuracy in prompting with context for both Gemini and Llama. Hence, through this analysis, we can state that both Gemini and Llama are accustomed to the terms within the range 0.096-0.062 despite them being the second most popular in the entire dataset. So, we can conclude that there is a lack of knowledge of the most popular terms across both models since we expect them to perform best for terms with the highest popularity. From this popularity metric, we can observe that the percentage accuracy has fluctuated in the distribution ranges, which showcases that we cannot trust this metric as a proper benchmark. This may have been because this benchmark is created solely based on our dataset and may not represent the importance of the terms as they are in the real world. On the other hand, the monthly page view metric consists of a varying range of terms starting from a massive 119,323 views per month to only 5 views per month before normalization. This is a better representation of the popularity of terms as the actual importance of terms on the internet can be seen using this metric. Therefore, for this metric, we

can see that there is a pattern in it, by the descent of popularity the accuracy has decreased. In both models, it is seen for both with and without context. Compared to Llama, Gemini has a better pattern as the performance decreases with a decrease in popularity but in Llama there are some portions where there is equal accuracy or a very small increase. However, for Llama the accuracy varied across different ranges so there is no definitive conclusion. Without context, Gemini has outperformed Llama with respect to accuracy until the popularity value 0.003. However, with the context provided to both models, Gemini performed better than Llama across all ranges. Hence, this proves that Gemini adapted better to the context in comparison to Llama and the normalized Wikipedia page view metric offers a more realistic pattern in results than the TF-IDF metric.

As we can see for both our metrics, the performances of both of our models are not very good for even the terms with the highest popularity, and providing context along with them increases the accuracy of the answers almost doubled. So, the adaptive retrieval method proposed by POPQA [16] is not applicable to the Bengali language even if it reduces latency. This is because in their research the models gave almost the same answer with or without context for the most popular terms. In order to achieve better and more proper answers, we will always need retrieval augmented generation for Bengali question answering until our models are capable enough to perform well for at least the highly popular terms.

## 6.2 Masked Token Prediction and its Analysis

After constructing the dataset as explained in section 4.3, we prompted the large language models to generate tokens for the [MASK] area for masked context, and we considered the top five predictions from each model. From the 5 predictions, the answers given by the large language models were considered accurate only if the term which is the actual answer is present as a substring of any of the generated predictions. For masked prediction, we considered the answer accurate only if the term is a substring of the actual answer since we want to determine how the large language models performed for that particular term that is present in its respective category. We used two monolingual models SahajBERT and BanglaBERT and two multilingual models Gemini and Llama. The accuracy of the models is demonstrated in table 6.1 below where Gemini performed best with an accuracy of 39.14% followed by SahajBERT with 25.37%, Llama with 13.29%, and BanglaBERT with 9.45%.

| Model | Accuracy |
|---|---|
| SahajBERT | 25.37% |
| BanglaBERT | 9.45% |
| Gemini | 39.14% |
| Llama | 13.29% |

Table 6.1: Overall Accuracy of Models for Masked Prediction

For masked prediction, we got the five outcomes in descending order of the probability of the outcome being the correct answer. Out of the 5 responses generated by the models we found out the answer detection quality by seeing how early the right answer appeared among the predicted ones and for that, we calculated the **Mean Reciprocal Rank (MRR)** for each model. Further discussion on masked prediction will be shown on the result of MRR for models.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i} \tag{6.1}$$

| Model | MRR Value |
|---|---|
| SahajBERT | 0.1663 |
| BanglaBERT | 0.0668 |
| Gemini | 0.3161 |
| Llama | 0.0815 |

Table 6.2: Overall MRR Values of Models for Masked Prediction

The overall MRR values for each of our models are shown in Table 6.2 where Gemini performed best (0.3161), followed by SahajBERT (0.1663), Llama (0.0815), and BanglaBERT (0.0668). We have further analyzed the MRR value for each category in our dataset as mentioned in section 4.4. This will help us to identify the performance of 4 models across each term and help us to identify in which aspect they are lacking knowledge.

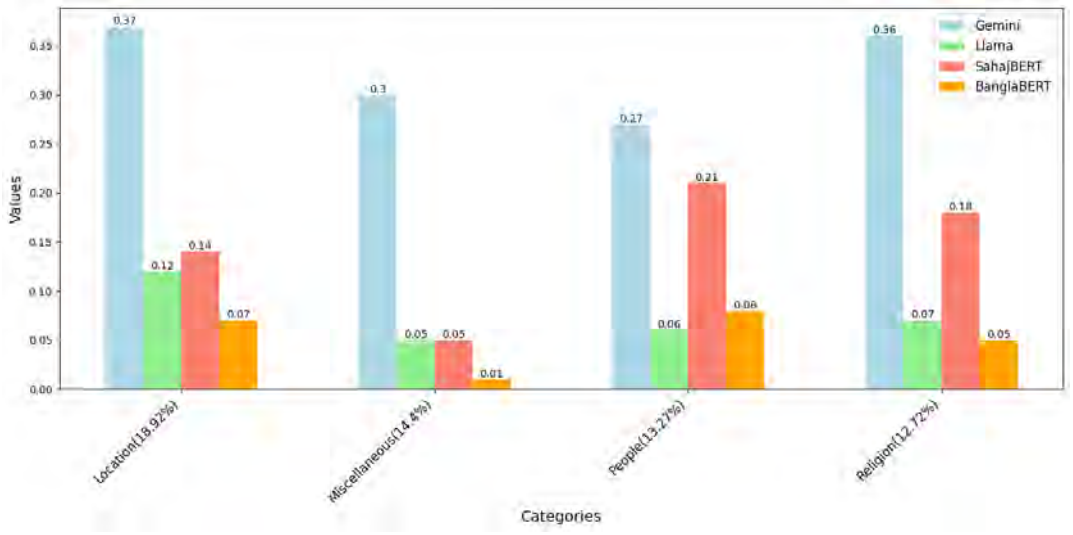


Figure 6.5: Bar Chart representing MRR Values of the top four categories

The figure 6.5 shows the comparison of MRR values of the top 4 categories (Location, Miscellaneous, People, Religion). Here Gemini has got the highest MRR value in all of the four categories. After Gemini, ShahajBERT performed the best among other models. As SahajBERT is trained on Bengali Wikipedia and our masked contexts were also generated from Bengali Wikipedia; the source being the same

makes it evident that the model will work better. BanglaBert performed worst in every category except the 'People' category, which is composed of names of famous individuals. For the same category, BanglaBERT outperformed Llama by 0.02. Gemini performed its lowest in this category whereas SahajBERT and BanglaBERT performed their best. Also, the performances of Gemini and SahajBERT are close in this category where they differ by 0.06. SahajBERT outperformed Llama in every category except 'Miscellaneous', where they both had the same MRR value.

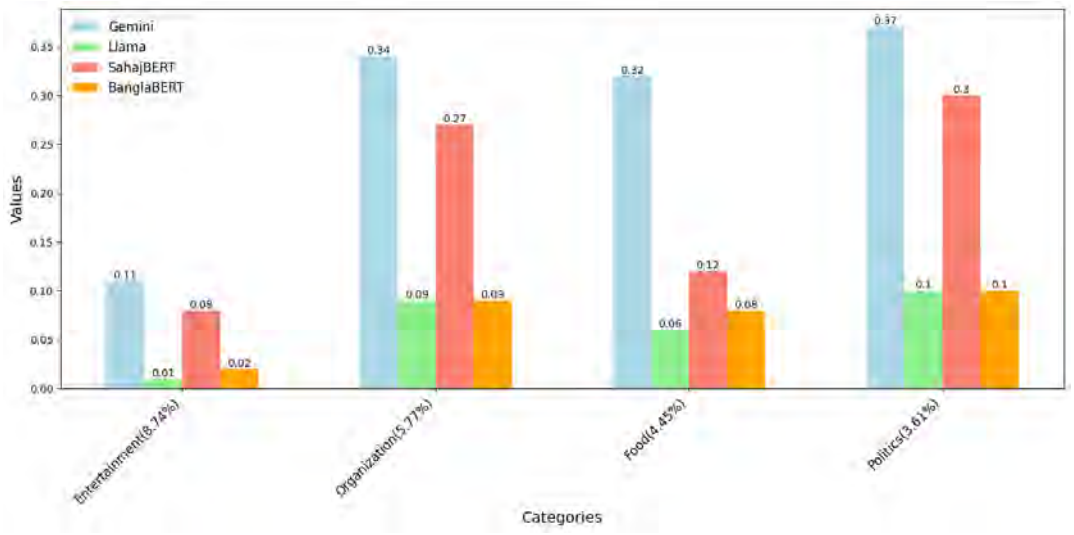


Figure 6.6: Bar Chart representing MRR Values for second top four categories

The figure 6.6 demonstrates the comparison of MRR values of the second top 4 categories (Entertainment, Organization, Food, Politics). Similar to the previous analysis Gemini has performed better than other models for all the categories. However, the performance for the 'Entertainment' category was low for all the models. SahajBERT has also performed significantly better than Llama and BanglaBERT. BanglaBERT performed similarly to Llama but in the case of the 'Entertainment' category, BanglaBERT outperformed Llama by a small margin of 0.01. Also for the 'Food' category, the MRR value of BanglaBERT is 0.02 more than Llama. With respect to the 'Organization' and 'Entertainment' categories, both models had equal MRR values. All the models performed poorly in the Entertainment category in comparison to other categories.

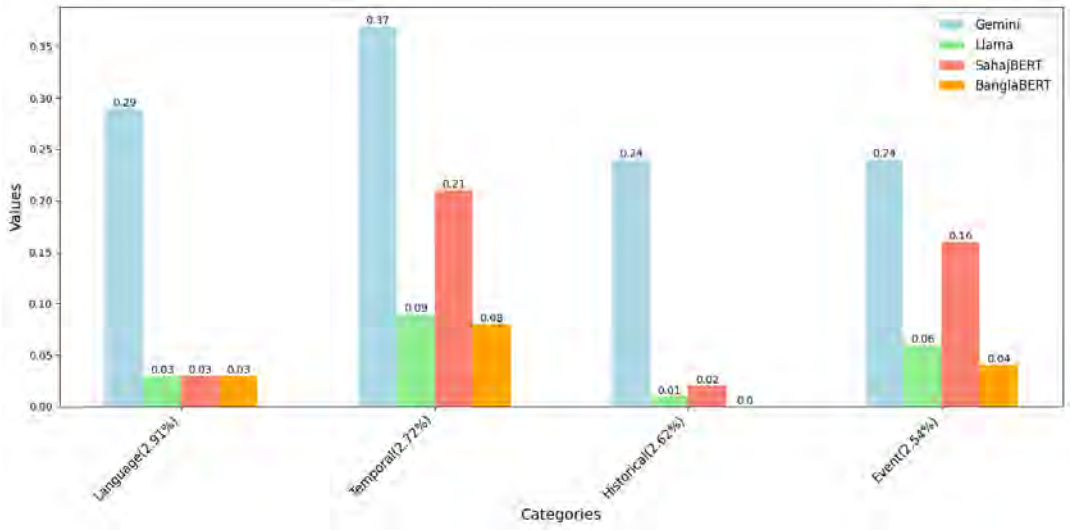


Figure 6.7: Bar Chart representing MRR Values for the second last four categories

This figure 6.7 above shows the MRR values of the second last four categories (Language, Temporal, Historical, Event). Just like previous charts, Gemini outperformed every model in every category. SahajBERT is the second best performing model among these models except in the 'Language' category. After Gemini and SahajBERT, Llama performed well in these categories. Llama even performed equally as SahajBERT in the 'Language' category. BanglaBERT performed well in every category except the 'Historical' category where its MRR value is 0.0. The reason behind this can be the lack of knowledge related to Bengali history in the model. In 'Language', it performed the same as Llama and SahajBERT. However, all the models performed significantly worse for the 'Historical' category when compared to the other 3 categories.

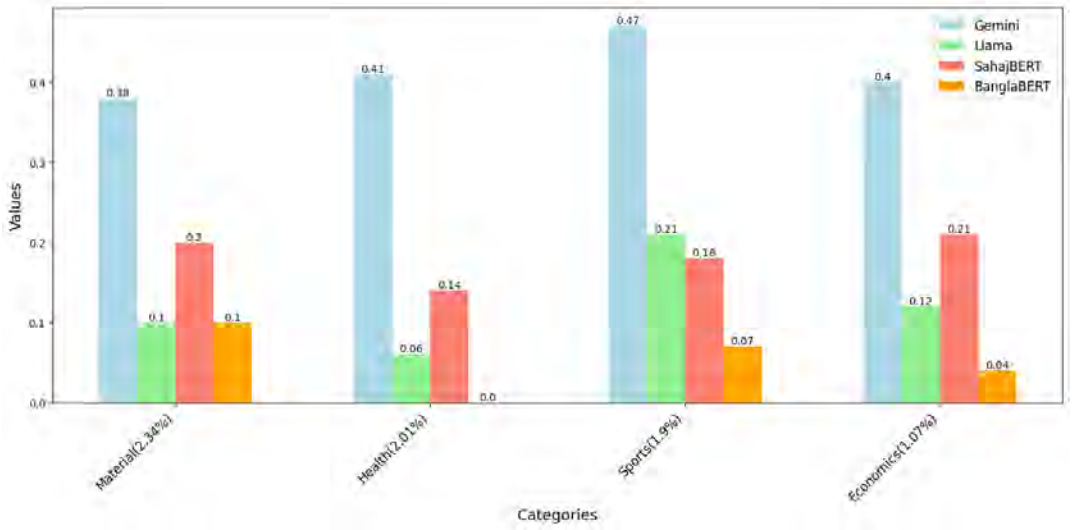


Figure 6.8: Bar Chart representing MRR Values for the last four categories

The figure 6.8 represents the MRR values of the last four categories (Material, Health, Sports, Economics). From the figure, it is evident Gemini has performed better than all the models for all 4 categories. SahajBert has outperformed Llama in

all categories except Sports where Llama outperformed SahajBERT by 0.03. Llama has overall performed better than BanglaBERT for these 4 categories except for the 'Material' category where both the models have the same MRR value. The MRR value of BanglaBERT for the 'Health' category is 0.0 which shows that BanglaBERT does not compose enough knowledge related to health.

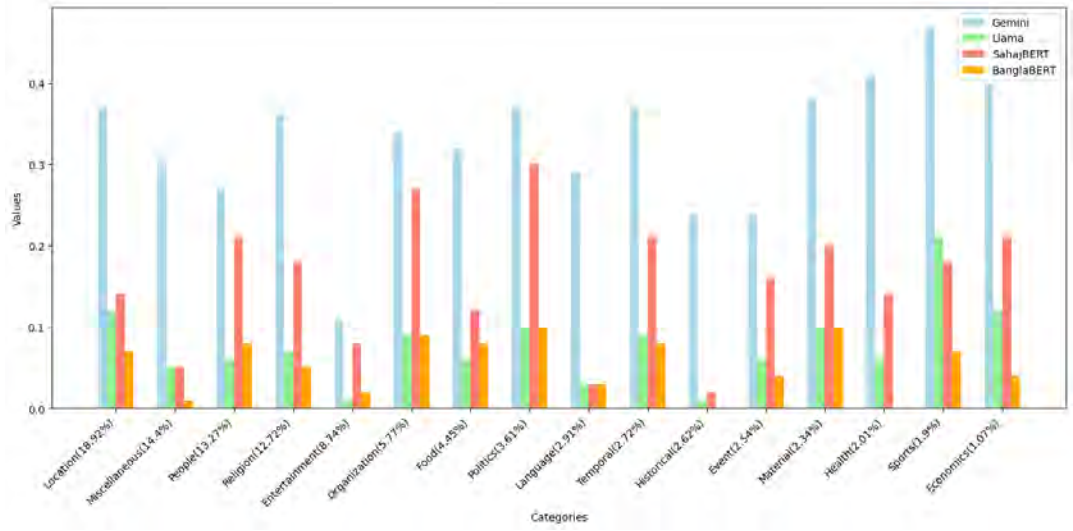


Figure 6.9: Bar Chart representing MRR Values for all categories

Figure 6.9 above portrays the sixteen categories and the performance of four models for each category. It shows that there is no fixed pattern; as some categories have a good percentage of terms but the results are poor and vice versa. It is also clear that whatever the category is, Gemini has performed best compared to the other three models. So, the performance of the model does not depend on the distribution of each category in our corpus; on the contrary it depends on the learning knowledge of the model.

## 6.3 Overall Analysis

### 6.3.1 Factual and Cultural Aspects

From the figure 6.9 we can see that 'Location', which has the highest percentage in our dataset, is a factual category and the models performed really well for this category. For the last four categories (Material, Health, Sports, Economics) their performance in masked prediction is comparatively better than other categories. 'Sports' category has the maximum value of MRR for every model in masked prediction testing. After this category 'Health', 'Material' and 'Economics' categories also have somewhat of a good grasp of knowledge but not as good as 'Sports'. Even in the 'Health' category, BanglaBERT has totally missed predicting almost every prompt correctly. Even though these categories have the least amount of terms in the dataset, their result score is quite impressive than other categories. Explanation of this result lies in the terms present in those categories as these four categories

consist of factual real life data. These data are factual and do not depend on culture which makes them available to every existing model in any language. For example, knowledge related to health does not depend on any culture and does not vary much from one part of the world to another. Same goes with economics as the economies of all countries in the world are interconnected and the patterns in economic conditions are similar. Therefore, we can say that when factual categories were prompted, all of the models could respond to them in a good manner and thus their performance is better than others.

On the other hand, observing the MRR values for categories like (People, Entertainment, History and Event) shows that the models performed poorly in these categories when compared to other categories in masked prediction.The 'Entertainment' category has the minimum MRR value for every model in masked prediction. The categories 'People', 'Historical' and 'Event' also show poor performances of the models. Furthermore, in the 'Historical' class, BanglaBERT was unable to predict any of the terms accurately, which indicates the lack of knowledge of BanglaBERT towards our historical data. Some arguments can be made where the poor performance of the model may be due to categories 'Historical' and 'Event' occupying least distribution in the dataset but this cannot be the reason as previous analysis shows exceptional performance for categories with low percentage. Even though categories like 'Entertainment' and 'People' occupy a significant amount of the dataset, the model's MRR value is relatively low. The reason for the model's failure may be due to the fact that the categories discussed are related to Bengali culture. The 'People' category comprises our politicians, writers, singers and many more. As most of the contributions made by these people lie on the cultural part, the poor performance of the large language models can be described as their lack of Bengali cultural knowledge. Coherently, entertainment, events and history are deeply intertwined with our culture, and the models are not accustomed with Bengali cultural terms and traditions, they were unable to perform efficiently. Therefore, from the analysis there is a lack of knowledge in large language models with respect to Bengali culture and traditions.

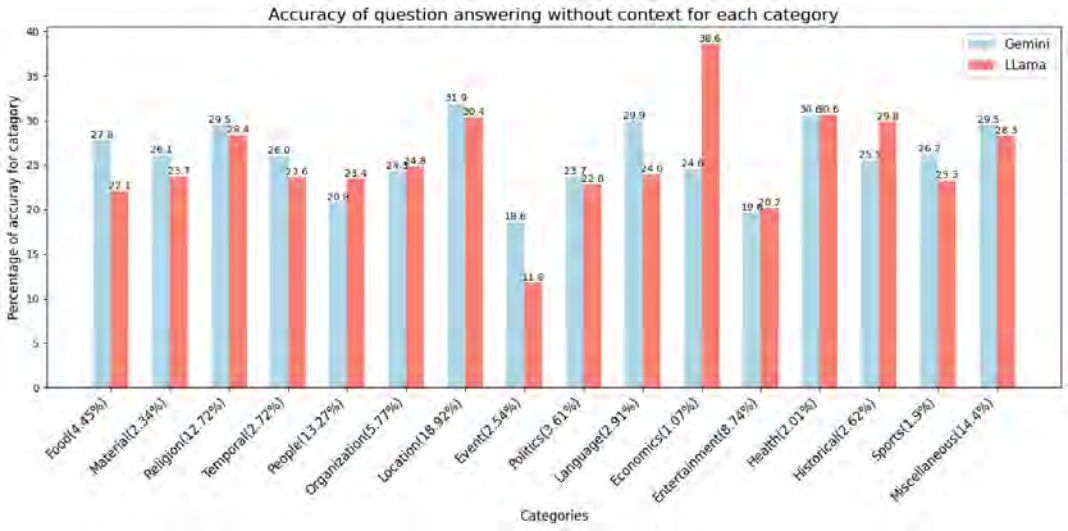


Figure 6.10: Accuracy for Question Answering across all categories

For checking the effect of factual and cultural aspects in question answering we will only see the question answering without context as with context, there is not much of a difference in result for distribution by popularity. From the figure 6.10, we can see the accuracy of the Gemini model for some of the cultural categories: for 'People' accuracy is 20.8%, for 'Entertainment' accuracy is 19.6%, for 'Politics' accuracy is 23.7% and for 'Event' accuracy is 18.6% and for some of the factual categories: for 'Material' accuracy is 26.1%, for 'Health' accuracy is 30.6%, for 'Sports' accuracy is 26.2%, for 'Economics' accuracy is 24.6% and for 'Location' accuracy is 31.9%. It is quite evident that in answering without context, the accuracy is favoring the factual categories irrespective of term's collection size. In the same way if we look into model Llama for those same categories we find that for factual categories: for 'Material' accuracy is 23.7%, for 'Health' accuracy is 30.6%, for 'Sports' accuracy is 23.3%, for 'Economics' accuracy is 38.6% and for 'Location' accuracy is 30.4% and for some of the cultural categories: for 'People' accuracy is 23.4%, for 'Entertainment' accuracy is 20.4%, for 'Politics' accuracy is 22.8% and for 'Event' accuracy is 11.8%. Here the observation is the same as before, as the 'Politics' category comprises political terms and names of uprisings in our region. Among the cultural categories, the 'Historical' category outperformed the 'Politics' category in question answering. Factual categories got better accuracy in both models compared to cultural categories. So even for questions answering, the observation of lack of knowledge of large language models towards Bengali culture is the same as we observed for mask prediction.

From the above analysis, we can come to the conclusion that when a model is prompted without context, the models work better for factual information compared to cultural information. Cultural knowledge is somewhat related to commonsense knowledge. As these models have greater expertise in factual knowledge than cultural commonsense knowledge, the results will be poor for cultural queries. So, in order to produce better results from these models we have to feed them data related to the Bengali culture, specifically focusing on areas such as information related to famous people, politics, events, entertainment, and history.

### 6.3.2 Model Performance

The table 6.2 and the figure 6.9 indicates that SahajBERT has performed significantly better than BanglaBERT for the task of masked token prediction. As discussed in section 5.1.1, SahajBERT is pre-trained on Wikipedia and OSCAR. It is also pre trained on natural language processing tasks like masked prediction so it is better suited for nlp tasks like masked prediction Since our context is also extracted from Wikipedia, so SahajBERT performed better in masked prediction. As explained in section 5.1.3, BanglaBERT is pre-trained on a large corpus, hence there was more noise in the dataset.Even if the model had the correct answer trained in it due to the noise it could not get the correct answer. This led to BanglaBERT resulting in overall poor performance in masked prediction compared to SahajBERT.

From the same tables it is evident that from the two multilingual models, Gemini

has performed drastically better than Llama with respect to masked prediction. Though Llama has performed really well in question answering but the performance degraded in masked prediction drastically. The drastic fall of Llama's performance is largely influenced by Llama's architecture. As explained in section 5.2.2, Llama is an auto-regressive model which means that it is suitable to predict the next component or prediction from previous input. Hence, in masked prediction it cannot accurately predict the [MASK] since with the help of only the previous tokens as input it is not able to provide an appropriate answer. In conclusion, Gemini performed best in both masked prediction and question answering. The performance of Llama is much better in question answering than masked prediction where in some categories and popularity ranges its performance was also slightly better than Gemini.

# Chapter 7

# Conclusion

Large language models are pretrained on a big corpus for many languages. Since large language models are pretrained on books, articles, and websites which often contain more factual information than cultural context, the models tend to concentrate more on factual information and ignore the cultural context. This imbalance in the training process results in a lack of cultural commonsense based knowledge in the large language model. Therefore, we tried to figure out the misbalance for the Bengali language in the large language model in both multilingual and monolingual models using the corpus generated by us. The dataset includes information that can be used for natural language tasks like question answering and masked prediction which helped us to analyze the reason behind the poor performance of large language models and to provide an analysis to determine which sectors of information the large language model is lacking in and in which sectors of information they are excelling. We found out that large language models have a good amount of knowledge regarding factual categories of information such as 'Sports', 'Economics', and 'Materials'. However, these models lack knowledge in categories that contain information related to the Bengali culture such as 'Events', 'Entertainment', and 'People'. Therefore, we can say that there is a lack of Bengali cultural commonsense knowledge in large language models. Moreover, our research found out that the accuracy of large language models increases with the increasing popularity of the topic, but Retrieval Augmented Generation will always be needed for Bengali as opposed to the English language until our models are improved at a certain level.

**Limitation:** We only used data from Wikipedia in our corpus which is composed of mostly formal language and information. Data from other formal sources was not considered in our research. Also, sources where informal language is used were not used to build our database. These sources include data from social media platforms such as Facebook, Twitter (X) and many more. The usage of social media for data extraction can be considered as a direction to our future work.

# Bibliography


[1] P. Singh *et al.*, "The public acquisition of commonsense knowledge," in *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, vol. 3, 2002.

[2] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. Li Zhu, "Open mind common sense: Knowledge acquisition from the general public," in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE: Confederated International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings*, Springer, 2002, pp. 1223–1237.

[3] J. Anacleto, H. Lieberman, M. Tsutsumi, *et al.*, "Can common sense uncover cultural differences in computer applications?" In *Artificial Intelligence in Theory and Practice: IFIP 19th World Computer Congress, TC 12: IFIP AI 2006 Stream, August 21–24, 2006, Santiago, Chile 1*, Springer, 2006, pp. 1–10.

[4] C. Havasi, R. Speer, and J. Alonso, "Conceptnet 3: A flexible, multilingual semantic network for common sense knowledge," in *Recent advances in natural language processing*, 2007, pp. 27–29.

[5] R. Speer, C. Havasi, *et al.*, "Representing general relational knowledge in conceptnet 5.," in *LREC*, vol. 2012, 2012, pp. 3679–86.

[6] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," *arXiv preprint arXiv:1811.00937*, 2018.

[7] T. H. Trinh and Q. V. Le, "A simple method for commonsense reasoning," *arXiv preprint arXiv:1806.02847*, 2018.

[8] M. Kejriwal and K. Shen, "Do fine-tuned commonsense language models really generalize?" *arXiv preprint arXiv:2011.09159*, 2020.

[9] V. Shwartz, P. West, R. L. Bras, C. Bhagavatula, and Y. Choi, "Unsupervised commonsense question answering with self-talk," *arXiv preprint arXiv:2004.05483*, 2020.

[10] Y. Xu, C. Zhu, R. Xu, Y. Liu, M. Zeng, and X. Huang, "Fusing context into knowledge graph for commonsense question answering," *arXiv preprint arXiv:2012.04808*, 2020.

[11] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar, "Indicbart: A pre-trained model for natural language generation of indic languages," *arXiv preprint arXiv:2109.02903*, 2021.

[12] K. Ma, F. Ilievski, J. Francis, S. Ozaki, E. Nyberg, and A. Oltramari, "Exploring strategies for generalizable commonsense reasoning with pre-trained models," *arXiv preprint arXiv:2109.02837*, 2021.

[13] Y. Sun, Q. Shi, L. Qi, and Y. Zhang, "Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering," *arXiv preprint arXiv:2112.02732*, 2021.

[14] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," *arXiv preprint arXiv:2104.06378*, 2021.

[15] M. Kowsher, A. A. Sami, N. J. Prottasha, M. S. Arefin, P. K. Dhar, and T. Koshiba, "Bangla-bert: Transformer-based efficient model for transfer learning and language understanding," *IEEE Access*, vol. 10, pp. 91 855–91 870, 2022.

[16] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," *arXiv preprint arXiv:2212.10511*, 2022.

[17] B. Muller. "Bert 101 - state of the art nlp model explained." (2022), [Online]. Available: https://huggingface.co/blog/bert-101.

[18] S. S. S. Jahan, R. Rab, P. Dutta, *et al.*, "Deep learning based misogynistic bangla text identification from social media," *Computing and Informatics*, vol. 42, no. 4, pp. 993–1012, 2023.

[19] T. Naous, M. J. Ryan, A. Ritter, and W. Xu, "Having beer after prayer? measuring cultural bias in large language models," *arXiv preprint arXiv:2305.14456*, 2023.

[20] M. AI, *Llama 3 model card*, https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md, Accessed: 2024-10-17, 2024.

[21] Cohere For AI, *C4ai-command-r-plus (revision 432fac1)*, 2024. DOI: 10.57967/hf/3138. [Online]. Available: https://huggingface.co/CohereForAI/c4ai-command-r-plus.

[22] M. Reid, N. Savinov, D. Teplyashin, *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[23] V. Yaadav, *Exploring and building the llama 3 architecture: A deep dive into components, coding, and inference techniques*, https://medium.com/@vi.ai_/exploring-and-building-the-llama-3-architecture-a-deep-dive-into-components-coding-and-43d4097cfbbb, Accessed: 2024-10-17, Apr. 2024.