Performance Comparison of Transformer-based Models for Multi-Reasoning in Machine Reading Comprehension

by

Sadika Sayma
20101131
Fariha Hasan Tonima
23341078
Sourav Biswas
20101324
Jannatul Ferdos
23341067
Tasnuva Haque
24141267

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

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:

Sadika

Sadika Sayma
20101131

Fariha Hasan Tonima
23341078

Sourav Biswas
20101324

Jannatul Ferdos
23341067

Tasnuva Haque

Tasnuva Haque
24141267

Approval

The thesis/project titled "Performance Comparison of Transformer-based Models for Multi-Reasoning in Machine Reading Comprehension" submitted by

1. Sadika Sayma (20101131)
2. Fariha Hasan Tonima (23341078)
3. Sourav Biswas (20101324)
4. Jannatul Ferdos (23341067)
5. Tasnuva Haque (24141267)

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

Examining Committee:

Supervisor:
(Member)

Ms. Najeefa Nikhat Choudhury
Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

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

Head of Department:
(Chair)

Sadia Hamid Kazi,PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Machine Reading Comprehension (MRC) is an artificial intelligence task that examines a given passage or text and answers queries regarding it. The objective is to make an intelligent support system that has the ability to understand the contextual information of the passage and give correct answers for multi-reasoning questions, commonsense based questions and multiple-choice questions, etc. One of the main challenges faced by MRC models in commonsense based and multi-reasoning questions is the need for understanding and reasoning beyond explicit textual information. To enhance the capabilities of MRC systems in these areas, the research focuses on the comparative analysis of state-of-the-art transformer-based models including BERT, ALBERT, RoBERTa, DistilBERT, MobileBERT, and ELECTRA. Our investigation specifically targets the enhancement of commonsense reasoning within MRC frameworks. In regards to this, we have used a binary decision making approach in our algorithm, in order to achieve a better outcome from these transformer-based models. To evaluate the performance, the experiments were conducted using CosmosQA dataset, which consists of narrative-driven questions that necessitate commonsense understanding to resolve.

Keywords: Machine Reading Comprehension(MRC), artificial intelligence, contextual information, multi-reasoning questions, commonsense, transformer-based models, BERT, ALBERT, RoBERTa, DistilBERT, MobileBERT, ELECTRA, CosmosQA.

Acknowledgement

Firstly, all praise to the Almighty Allah for whom our thesis have been completed without any major interruption. Secondly, to our supervisor Ms.Najeefa Nikhat Choudhury ma'am for her kind support and advice in our work. She helped us whenever we needed help. And finally to our parents, without their throughout support it may have not been possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several abbreviation that will be later used within the body of the document

ALBERT	A Lite BERT
BERT	Bidirectional Encoder Representations from Transformers
CLS	Classification Token
DistilBERT	A Distilled Version of BERT
ELECTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurately
FN	False Negative
FP	False Positive
MCQ	Multiple Choice Question
ML	Machine Learning
MLM	Masked Language Modeling
MRC	Machine Reading Comprehension
NLP	Natural Language Processing
QA	Question Answering
RoBERTa	Robustly Optimized BERT
SEP	Separator Token
SOP	Sentence Order Prediction
TN	True Negative
TP	True Positive

Chapter 1

Introduction

Over the past few years, question-answering (QA) systems have made significant improvements in natural language processing (NLP) and are now an important feature of many applications such as virtual assistants, chatbots for customer service as well as data retrieval systems. Also by detecting the context and pulling relevant information from a huge text, QA systems are providing detailed and brief responses to the user inquiries. The management of multi-reasoning queries, when responding to it, requires combining data from several sources or utilizing various reasoning processes which is one of the most significant challenges encountered by the QA systems. As they mainly rely on the exact matching of words or predetermined rules, traditional QA methods like keyword matching or rule-based systems frequently struggle with such inquiries by making them less successful at capturing complicated interactions and nuanced reasoning. For NLP applications such as machine translation, language modeling, and sentiment analysis, transformer-based algorithms have become a strong answer. Transformers, introduced by Vaswani et al. (2017)[7], have transformed the area of NLP by utilizing the parallelization approaches and self-attention processes. These models such as BERT (Bidirectional Encoder Representations from Transformers) [10], ALBERT (A Lite BERT) [16], RoBERTa (Robustly Optimized BERT) [13], DistilBERT (A distilled version of BERT) [18], MobileBERT [19] and ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) [15] have established themselves as the standard practice in many NLP applications due to their outstanding performance across multiple benchmark datasets.

In this paper, we will be focusing on multi-reasoning and commonsense based question answers using transformer-based algorithms to evaluate the performance by conducting a few experiments on the CosmosQA[11] dataset. Our goal is to develop a robust and efficient QA system that can handle complex queries and provide accurate answers by effectively incorporating multiple reasoning processes. We aim to explore and enhance the capabilities of transformer models to improve their performance on multi-reasoning tasks. The outcomes of this research can have significant possibilities in a variety of categories like education, healthcare, and information retrieval, where these effective QA systems can greatly enhance the user experiences and provide valuable insights from data in text.

1.1 Problem Statement

Machine Reading Comprehension (MRC) is a great accomplishment in natural language processing (NLP) which assists in extracting the desired answer or response from a given text. Despite considerable improvements over the past few years, the lacking in this field of work is quite significant; mainly in regards to the effective extraction of data out of textual information.

The main hindrance in establishing the task is comprehending the accurate and precise inner meaning and representation of a complex textual representation. Ordinary MRC models struggle to understand context-dependent meaning and structure behind the natural language text. Thus, where a deep understanding is needed, the performance and the analysis desirability degrades, and the credibility of getting accurate answers to the questions also falls off.

Secondly, the generalization of MRC tasks is a mandatory part while answering the multi-reasoning questions. Running the models on very few ordinary, unreliable datasets leads the model to become very much unadaptable while facing new structural texts and results in inappropriate, undesirable answering.

Thus, there is a need for much more advanced, interpretable, and explainable MRC models. Ordinary models lack in exploring the inner meaning perfectly, making it difficult to see how the decisions are made to arrive at the solutions. Accountability is an unavoidable part of MRC. Crystal clear and accurate interpretable answers/explanations are required in areas such as law or medical fields, in order to gain trust. Furthermore, biases in the datasets can also be an issue. That can result in answering biased or unfair predictions and so on. So, working with reliable and many datasets to get rid of biases has to be another indispensable concern. Therefore, This study proposes novel methodologies to overcome the MRC's aforementioned problems to enhance accuracy while answering the multi-reasoning or commonsense related questions. By applying some great transformer-based algorithms like BERT, ALBERT (A Lite BERT), RoBERTa, DistilBERT, MobileBERT and ELECTRA , this research ensures that the best interpretability comparing these models.

1.2 Research Objective

The goal of this study is to evaluate and compare different transformer-based models to see how well they improve the abilities of machine reading comprehension (MRC) systems, especially in terms of multi-reasoning and commonsense reasoning. By working with the CosmosQA dataset, which focuses on narrative-driven, commonsense-based questions, we aim to identify which model performs best in this dataset. Ultimately, we want to create a reliable, intelligent support system that can understand and process complex information accurately, enhancing the overall effectiveness of MRC models.The main objectives of our research are given below:

1. Improve the accuracy of commonsense based QA task.
2. Compare and analyze the performance of different transformer-based models on commonsense reasoning QA tasks.
3. Measure each model's performance using accuracy.
4. Evaluate the effectiveness of transformer-based models in handling multi-reasoning and commonsense based questions.

Chapter 2

Literature Review

Transformer-based algorithms have emerged as useful tools in natural language processing tasks in recent years.The fundamental justification for employing transformer-based algorithms is that their attention mechanisms and contextual embeddings have shown exceptional performance in a variety of NLP applications. We came across some articles showing intriguing findings while researching MRC using transformer-based algorithms. In this section, we go over some earlier research that employed transfer learning to complete problems involving machine comprehension of text.

Natural Language Processing(NLP) and commonsense reasoning have come together to create various complex models that can understand human narratives beyond the actual meaning. Among the pioneering endeavors in this field, Huang et al. (2019)[11] introduced CosmosQA which is a large-scale dataset aimed at advancing machine comprehension through contextual commonsense reasoning. The dataset has 35,588 multiple-choice questions derived from 21,886 unique paragraphs. Additionally, the dataset requires models to evaluate assumed narratives, hypothetical scenario and the likely causes and effects within a given environment. The authors experimented this benchmark dataset with different state-of-the-art neural architectures including their novel BERT with Multiway Attention model. Despite the authors used advanced approaches, the research has a significant gap between human and machine performance. The highest score the model performed is 68.4% and the human performance is 94% which indicates the path for future exploration in the realm of commonsense machine comprehension. This gap not only draws attention to the difficulties in combining reading comprehension with common sense reasoning, but it also opens up opportunities for research on artificial intelligence (AI) with the goal of bridging the complex comprehension that humans naturally acquire.

A novel dataset designed by Talmor, A., Herzig, J., Lourie, N., and Berant, J. (2019)[8] indicates focus on commonsense knowledge for answering questions for specific contextual information.The CommonsenseQA dataset consists of 12,247 multiple-choice questions which need commonsense reasoning for accurate resolution. The main challenges the authors mentioned is the generation of the questions that not only requires commonsense reasoning but also the level of difficulty that is non-trivial for both traditional and advanced models including transformer based models. The human accuracy on this dataset is 88.9%.The most accurate baselines are BERT-LARGE and GPT, with respective accuracy of 45.5% and 55.9% on the random split (63.6% and 55.5% on the question concept split). This shows how much easier the benchmark is for humans because it is significantly below human accuracy. This finding is significantly greater than chance (20%) and indicates that language models are capable of storing substantial amounts of data pertaining to commonsense knowledge.

The paper that was released by the U.khanna and D.Mollá(2021)[22], experiments in the BioASQ Task 9b Phase B challenge, specifically focusing on biomedical question answering which requires multi-reasoning. They investigated the use of transformer-based language models and techniques. Acknowledging the importance of fine-tuning they stated that the fine-tuning approaches are important for learning distributions of the target task and improving the language model's adaptability. There are many challenges of fine-tuning on small datasets like BioASQ because it can cause catastrophic forgetting and overfitting. Catastrophic forgetting refers to the phenomenon where the model forgets what it learned previously which causes performance drop when trained on a new task. Catastrophic forgetting is problematic in scenarios where the model needs to learn continuously to adapt to new tasks while retaining knowledge from previous tasks. To overcome these challenges they explored scheduling like gradual unfreezing, where model layers are fine-tuned gradually instead of all at once. Based on the ALBERT and DistillBERT models they described their two systems. At first they employed a staged fine-tuning on the SQuAD2.0 dataset and then on the BioASQ9b dataset. The DistilBERT-based system examines how a more compact, smaller transformer-based model responds to gradual unfreezing. The paper provides insights into the pre-processing procedures where it is needed to convert the BioASQ dataset into the SQuAD format for training and evaluation. The factoids are considered as extractive QA tasks, with the answers being taken directly from relevant snippets. They used Strict Accuracy (SAcc), Lenient Accuracy (LAcc), and Mean Reciprocal Rank (MRR) evaluation matrices. Their experimental results show that the ALBERT-based systems ranked first in test batch 1 and fourth in test batch 2 for factoid questions. Surprisingly, the DistilBERT systems outperformed the ALBERT variants in test batches 4 and 5, despite having significantly fewer parameters which was 81% fewer. The systems "DistilBERT" and "Unfreezing DistilBERT" have average MRR scores of 0.5209 and 0.5232 for test batches 2, 4, and 5, respectively, and the disparity is not statistically significant. The accuracy of the model was not significantly affected by gradually unfreezing in comparison to conventional fine-tuning, according to the authors.

K. Pearce, T. Zhan, A. Komanduri and J.Zhan (2021)[23], conducted a comparative study of transformer-based language models on extractive question answering. They aim to evaluate the generalizability of these models across different datasets with diverse complexities, while also introducing a new model architecture called BERT-BiLSTM for extractive question answering to examine if adding bidirectionality can improve the model performance. Extractive question answering involves taking a section of text from a provided context paragraph and using it as the response to a given question. They described the integration of BERT [10] as an encoder layer for tokenization which captures contextual information from the input text and uses the self attention transformer to provide a deeper encoding than earlier embedding models like Word2Vec[2] or Glove [4]. They discovered that BERT's bidirectionality makes it significantly more effective as an encoder than merely LSTM[1]. In order to improve the model's comprehension and representation of the context needed to respond to the queries, the authors also added a BiLSTM layer. They expressed the input sequence into the language models as a question context single packed sequence in order to augment the question and context together. In the experiment part, four separate datasets which are NewsQA, SQuAD 2.0, QuAC, and CovidQA are used to fine-tune a number of pre-trained language models, including XLNet, BERT, RoBERTa, ALBERT, ConvBert, and BART. They emphasized the distinctive features of each dataset, including open-ended questions, complicated reasoning requirements, dialogue exchanges, and unanswerable queries. Their findings demonstrated that the models worked best on the SQuad 2.0 dataset, which contained simple contexts, questions, and extracting replies. On the NewsQA dataset, they did reasonably well, showcasing their outstanding reasoning abilities. On the QuAC dataset, which was made up of open-ended questions requiring inference, the models did badly, nevertheless. Due to the small amount of training data and the short maximum sequence length restriction, the CovidQA dataset which contains longer contexts, questions and responses, created a difficulty for the models. RoBERTa and BART had the best performances out of all the models. Their suggested model BERT-BiLSTM, performed 1% better than BERT basic. Therefore, adding a bidirectional LSTM layer improves contextual representations and improves performance in question-answering, claims their paper.

A model and data collection called WebSRC were created by Lu Chen et al.[20] for answering questions about websites. Their main goal was to effectively highlight the challenges of understanding web page content and emphasizes the importance of considering both text and structural information. For instance, if you are given a query and a web page, the goal is to find the answer on that website. Their proposed dataset WebSRC is a novel Web-based dataset for structural reading comprehension that includes 400k question-answer pairs that were gathered from 6.4K web pages. For each web page, the dataset contains HTML code, screenshots, and information. Each segment's questions were carefully crafted by the authors, and the replies were either text extracts from the web page or yes/no responses. The increased size and inclusion of HTML documents and images of the WebSRC dataset are highlighted in comparison to other datasets that contain HTML documents. The experimental setup involves training baselines on the dataset and evaluating their performance. Comparisons were performed between baselines using different combinations of text, HTML tags and screenshots. The findings demonstrate that including more context information enhances performance, with models using HTML components and screenshots outperforming models that only use text. They reported the result of SQuAD model on their dataset without fine-tuning. Therefore the exact match score is only 29.68 which means the texts from HTML are highly different from the normal textual passages. The authors realized the need for more sophisticated technology to model the HTML structure because, although virtually matching the performance of the version without pretraining in all metrics, the fine-tuned models still perform quite differently in the textual QA dataset. The improved performance of ELECTRA-based models is seen in the comparison of BERT-based and ELECTRA-based models, highlighting the difficulties the WebSRC dataset presents for pre-trained language models at the moment. They highlighted the necessity of future research into the use of structural information and emphasized the significance of combining layout features with textual contents for web page understanding.

The authors Hu Xu, Bing Liu, Lei Shua, and Philip S. Yu(2019)[14] provided a thorough analysis of the performance of BERT in review-based tasks because they recognize the value of question answering in e-commerce as it gives customers the opportunity to actively seek out crucial information about products or services in order to aid their purchase decision. Their main objective is to transform customer reviews into a significant knowledge base that can be used to address customer questions. They explored the effectiveness of post-training BERT on tasks such as Review Reading Comprehension (RRC) and Aspect-based Sentiment Analysis (ASA). Highlighting the challenges and significance of understanding review-based tasks in natural language, the authors identified three research questions(RQs) from the basis of their experiments. These RQs revolve around the performance gain achieved by BERT post-training, the performance of BERT's pre-trained weights without domain and task adaptation and the individual contributions of domain knowledge post-training and task-awareness post-training to the overall performance gain. They conducted their experiments for RRC, AE, and ASC tasks using existing datasets such as SemEval and SQuAD[6]. To guarantee high-quality data, they used an exacting annotation method.For post-training domain knowledge, they use Yelp Dataset Challenge and Amazon laptop reviews[5], and SQuAD 1.1 for post-training task awareness. The researchers compare various BERT models as well as current state-of-the-art models for AE and ASC tasks. BERT-DK (post-training on domain knowledge), BERT-MRC (post-training on SQuAD 1.1), and BERT-PT (proposed joint post-training algorithm) are some of the variants. A baseline approach, DrQA, and DrQA+MRC (fine-tuning on ReviewRC with training weights) are also included. The results show that BERT post-training, specifically with the suggested joint post-training method (BERT-PT), outperforms state-of-the-art models and BERT's pretrained weights alone in RRC and ASA tasks. The authors emphasize how well BERT post-training works to enhance performance on review-based tasks. Their recognition of the contributions of task awareness and domain knowledge post-training adds complexity to the subject and has implications for further study.

When it comes to applying BERT-based models to process lengthy texts, Xueqiang Lv et al.(2021)[25]provide an innovative method. The impact of entirely natural language processing tasks has been elevated with the development of a large-scale pre-training model based on the transformer model. These models, however, struggle to absorb lengthy texts because of the great complexity of the transformer's self-attention mechanism. To efficiently manage lengthy texts, the authors suggested a technique dubbed HBert that combines the strength of BERT and hierarchical attention mechanisms. They claimed that techniques including truncation, segmentation, compression, and structural alterations had limitations when used to handle lengthy texts with BERT. The authors suggested using the segmentation-based HBert method to get over these drawbacks. The lengthy text is broken up into several sentences and in order to create sentence vectors, each sentence is BERT and word attention layer encoded. The sentence encoder then uses a transformer and sentence attention to retrieve the article vector. For further tasks like text classification and question answering, the produced article vector is employed. The outcomes of the experiments show how effective the HBert approach is. On a variety of open datasets, including as the WikiHop QA dataset[9], Hyperpartisan news[12], and IMDb movie reviews, it surpasses the state-of-the-art longformer model in both text classification and QA tasks. With 95.7% in longer text classification tasks and 75.2% in QA tasks, HBert's F1 scores are remarkable. In their paper the authors also included ablation experiments to evaluate the significance of various model constituents. By resolving the difficulties of capturing word-level and inter-sentence semantic information, these tests demonstrate that word attention and sentence attention significantly contribute to the model's performance. The research also analyzes the memory requirements of the HBert model and shows that it is less complex than the BERT and RoBERTa models.

Chapter 3

Description of the Models

3.1 Introduction to Transformer Model Architecture

A transformer is a neural network architecture which has the ability to capture highly complicated dependencies and relationships in any complex tasks. These models have parallelization which makes the computation more efficient for tasks having long-sequences. Thus, for question answering problems, transformer models are mostly used as they can facilitate better results.
The architecture of the transformer plays a crucial role for its enhanced performance compared to other architectures. Among the tasks it can perform, one is the question answering task. This is due to its ability to understand complex contextual information and how it can analyze as well as generate answers in respect to it. It can also handle long sequences quite efficiently. In order for this transformer to understand the sequence at first the dataset needs to be trained in the model. Therefore, to ensure this the dataset is firstly tokenized into tokens. The tokens are then passed over to the transformer layers where the training of the tokens takes place. The inputs of the models, the tokens, are separately created. One token is for the questions and the other for the contexts. To identify the tokens from one another a special token [SEP] also known as the separator token, which is added in front of the question and context sequences. Another type of special token is present [CLS] which is known as the classification token and this particular token is the first token to enter before the input sequence as this helps to understand the relationship between part of the data and in order to distinguish between the beginning of a new set of data.The fig. 3.1 shows the basic architecture of transformer.

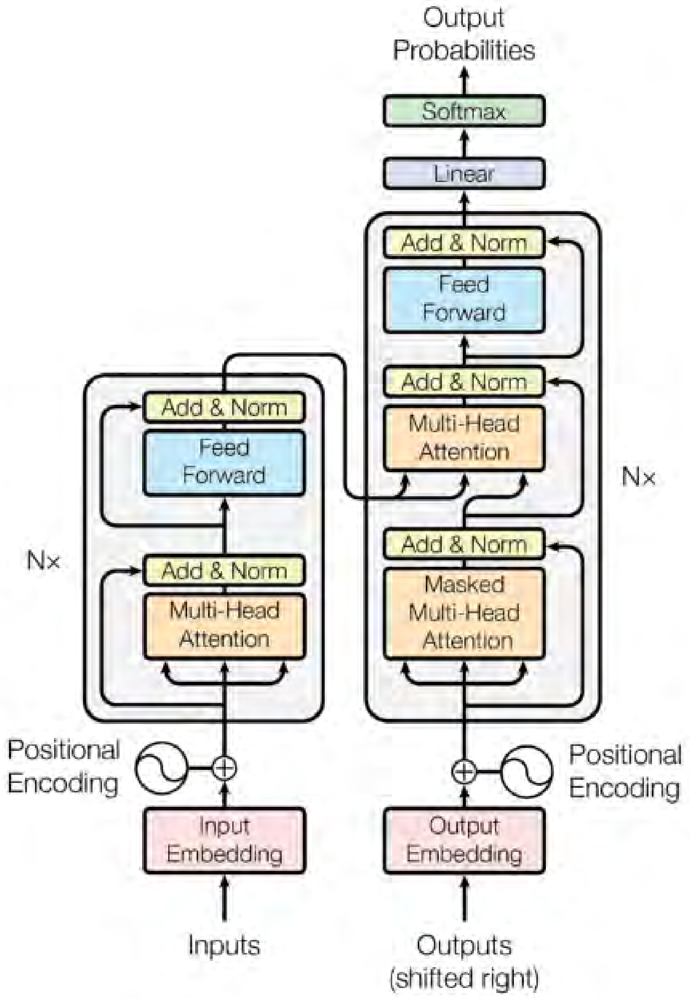


Figure 3.1: Basic Transformer Architecture [7]

The architecture of the transformer models in which the tokens are passed over has the following layers:

1. **Embedding Layer:** The sequence, having undergone tokenization for input, traverses through this stratum of the transformer architecture, facilitating the conversion of tokens into high-dimensional vectors.

2. **Encoder Layer:** Within this layer, the computation of positional encoding, which involves discerning the position of the tokenized data within the sequence, takes precedence. Additionally, this layer incorporates a dual-mechanism configuration. The initial mechanism involves a self-attention process, empowering the models to adeptly grasp the contextual nuances embedded within the input sequence. Subsequently, the subsequent layer features a feed-forward neural network, wherein non-linear transformations unfold to meticulously capture the intricate syntactic details existing between the input sequence and the interplay among the tokens.

3. **Decoder Layer:** In this layer, the encoded input sequence is passed over and then three mechanisms take place which are the masked multi-head self-attention, encoder-decoder attention and another feedforward networks. The

first one is responsible of applying mask to the above self-attention mechanism for preventing the positions to visit the next position. The next mechanism ensures to calculate an attention score between the encoder's output and the decoder layers input, finally combining the results for the input sequence. Lastly, there is a feed-forward neural network which works similar to the one mentioned in the encoder layer.

4. **Linear Layer:** The linear layer is also the outer layer of the model and here the ultimate hidden state of the [CLS] token undergoes a transformative process. This token possesses the remarkable capacity to comprehend the input sequence, having traversed through the encoder layer. In this transformation, the initially high-dimensional vector representing the [CLS] token is skillfully mapped to a vector of a size precisely matching the total count of distinct classes integral to the given classification task.

5. **Softmax Function:** The raw output values that we have received till now, will be converted to class probabilities by this function.The softmax function, as defined in Equation 3.1, is crucial for transforming logits into probabilities, which is fundamental in the classification tasks performed by the model.

$$\text{Softmax}(QK^T)_i = \frac{\exp((QK^T)_i)}{\sum_{j=1}^{N} \exp((QK^T)_j)} \tag{3.1}$$

6. **Prediction:** Prediction is the final stage here. This step will ensure the class possessing the utmost probability, obtained via the application of the softmax function, is designated as the input sequence instrumental in determining the ultimate prediction.

3.2 BERT

BERT(Bidirectional Encoder Representations from Transformers) is one of the central models in this investigation [10] . This model marks a pivotal advancement in the realm of natural language processing (NLP). Its strength lies in its capacity to comprehend contextual details. BERT is well renowned for its bidirectional technique. This technique enables the model to take into account both preceding and subsequent words within a sentence when deciphering word significance and context. One crucial part of BERT's training involves the Next Sentence Prediction (NSP) task. This task helps the model grasp the connection between consecutive sentences, thereby improving its ability to understand context. This helps BERT achieve a high level of expertise in grasping the subtleties of language, establishing it as a fundamental component in numerous NLP assignments. The fig. 3.2 below illustrates the architecture of BERT model.

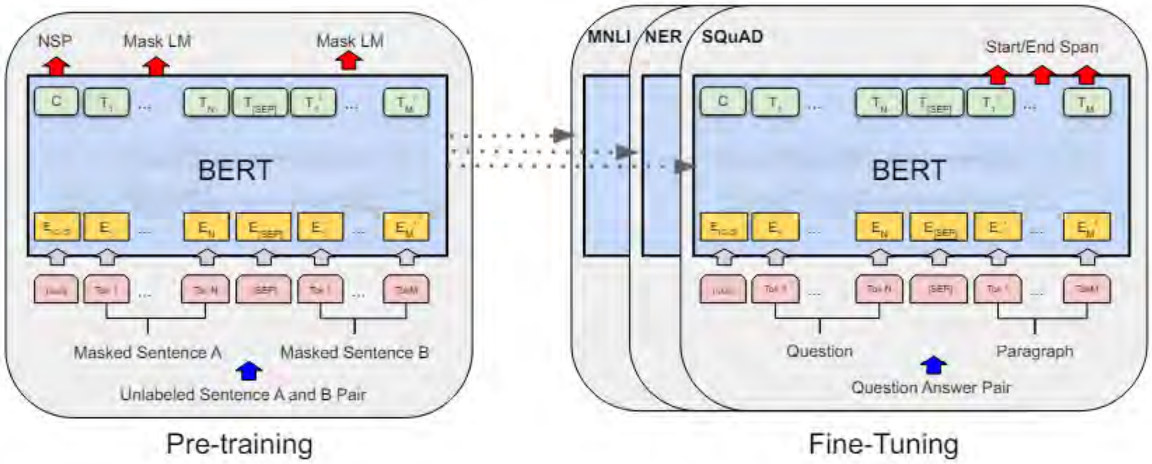


Figure 3.2: BERT Architecture [10]

3.2.1 Architecture of BERT

The structure of BERT comprises several Transformer layers. Each of the layers contains a multi-head self-attention mechanism and a feedforward neural network. The model BERT employs a bidirectional strategy, enabling it to grasp contextual information by taking into account both preceding and following words in a sentence [10].The bottom row in fig. 3.2 shows the input tokens `[CLS]`, `[SEP]`, Tok1, Tok2, . . . , TokN.The final hidden state of BERT model can be used as the aggregate sequence representation for classification tasks.The inputs are represented as a pair of sentences (A and B). Here, the sentence pairs are either consecutive sentences from a corpus to help learn sentence relationships or randomly paired sentences, which does not represent a direct relationship.The input tokens include special tokens such as $[CLS]$ (used at the beginning and for classification tasks), $[SEP]$ (used to separate the sentences) and tokens from sentences A and B. Each token in BERT model is at first converted into embedding vector.These vectors are shown as $E_{[\mathrm{CLS}]}, E_1, \ldots, E_N, E_{[\mathrm{SEP}]}, E_1', \ldots, E_M'$ in the fig. 3.2. During pre-training, the random tokens in the input are masked or replaced with a special $[MASK]$ token and the model learns to predict the original token based only on its context. The model predicts whether the second sentence in the pair follows the first sentence logically and sequentially. This helps BERT learn relationships between sentences. The central large block labeled “BERT” in the fig. 3.2 represents the multiple layers of the transformer model. Each layer consists of multiple self-attention heads and a feed-forward neural network. The transformer block processes the input embeddings and produces outputs that are fed into the next layer.The outputs of the transformer blocks for each token is shown as $T_1, \ldots, T_N, T_{[\mathrm{SEP}]}', T_1', \ldots, T_M'$ in the fig. 3.2.

These outputs are the transformed representations of each input token after considering the context provided by other tokens in the sequence.The input during fine-tuning is adapted to specific tasks.This includes pairs of a question and a paragraph for question answering tasks such as CosmosQA. The outputs of the BERT model are labeled accordingly depending on the task.For instance, in a question-answering task, the labels would indicate the start and end of the answer in the context paragraph. The transition in the fig. 3.2 indicates the transition from pre-training to fine-tuning.The pre-trained model which has learnt a rich representation of language features and relationships, is fine-tuned using a smaller amount of task-specific data to tailor its predictions to specific NLP tasks.

3.3 ALBERT (A Lite BERT)

ALBERT is developed based on the strengths of BERT while addressing some of its resource intensive characteristics[16]. It delivers similar performance to BERT while requiring significantly less computational resources. ALBERT is a great choice, especially when dealing with limited computational resources, without compromising on the quality of results. The fig. 3.3 shows the architecture of ALBERT model.

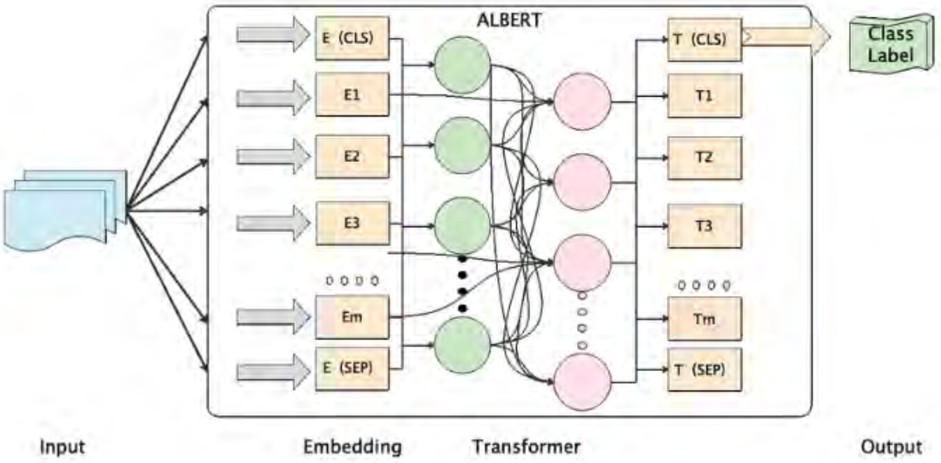


Figure 3.3: ALBERT Architecture [17]

3.3.1 Architecture of ALBERT

ALBERT uses two methods for parameter reduction to accomplish this. A factorized embeddings parameterization is the first method and the second method is shared cross layer method. In addition, ALBERT employs a self-supervised loss function for Sentence Order Prediction (SOP). The fundamental objective of the SOP (Sentence Order Prediction) is to enhance the coherence between sentences. It aims to rectify the shortcomings of the Next Sentence Prediction (NSP) loss introduced in the actual BERT model. [16].

3.4 RoBERTa

RoBERTa is a derivative model inspired by BERT[13]. This model has garnered recognition for its strong performance across various NLP applications. Its excellence is attained through extensive pre-training on extensive text collections and subsequent fine-tuning on a range of practical tasks. RoBERTa's flexibility and proficiency position it as a formidable contender for tackling intricate MRC difficulties.The fig. 3.4 shows the architecture of RoBERTa model.

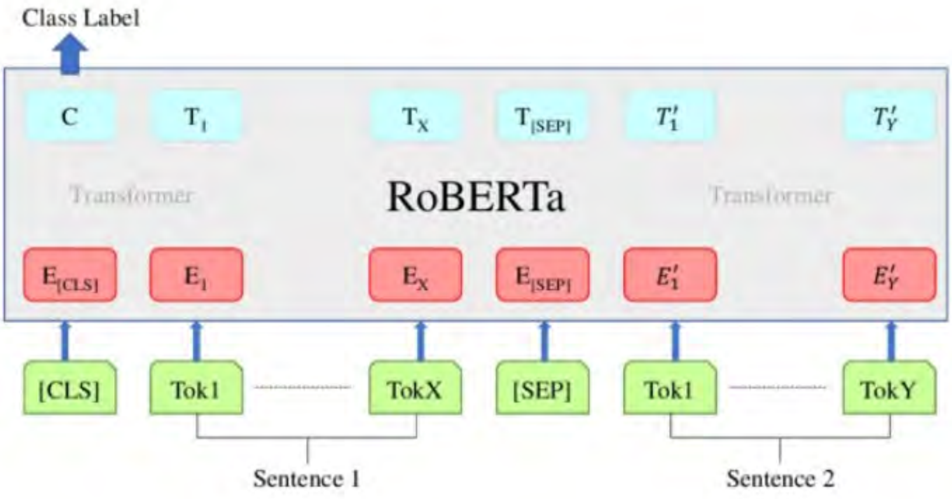


Figure 3.4: RoBERTa Architecture [26]

3.4.1 Architecture of RoBERTa

RoBERTa shares a similar resemblance to BERT in terms of architecture but places a strong emphasis on extensive pre-training[13]. It involves more extensive training on vast text collections and dispenses with BERT's next sentence prediction task, resulting in enhanced performance [13].In RoBERTa, the input consists of two special tokens which are `[CLS]` and `[SEP]`.Each of these tokens is converted into embedding vector.These embeddings ($E_{[\text{CLS}]}, E_1, \ldots, E_N, E_{[\text{SEP}]}, E'_1, \ldots, E'_Y$) in the fig. 3.4 serve as the input features for the transformer blocks. They encapsulate both the semantic meaning of the tokens and their positional information. RoBERTa processes the embeddings by the transformer blocks, which apply self-attention mechanisms by allowing the model to consider other tokens in the sentence while processing a specific token. This helps in understanding the context around each word. The fig. 3.4 shows intermediate token representations ($T_1, \ldots, T_x, T_{[\text{SEP}]}, T'_1, \ldots, T'_Y$) after processing through one or more transformer blocks.The top-most layer is either a linear or a softmax layer that takes the transformed [CLS] representation and outputs a class label. This label can represent categories like sentiment, entailment, agreement, depending on the specific task.

3.5 MobileBERT

MobileBERT is particularly designed for resource limited environments such as mobile devices and it is also a lite version of BERT model [19]. This model not only reduces the size and the computational units but also maintains a high level performance of larger models like BERT. Therefore, the mobileBERT model is an appropriate choice for the devices or applications that need advanced Natural Reading

Comprehension(NLP) abilities but has limited hardware capabilities such as smartphones and other portable devices.The fig. 3.5 shows the architecture of MobileBERT model.

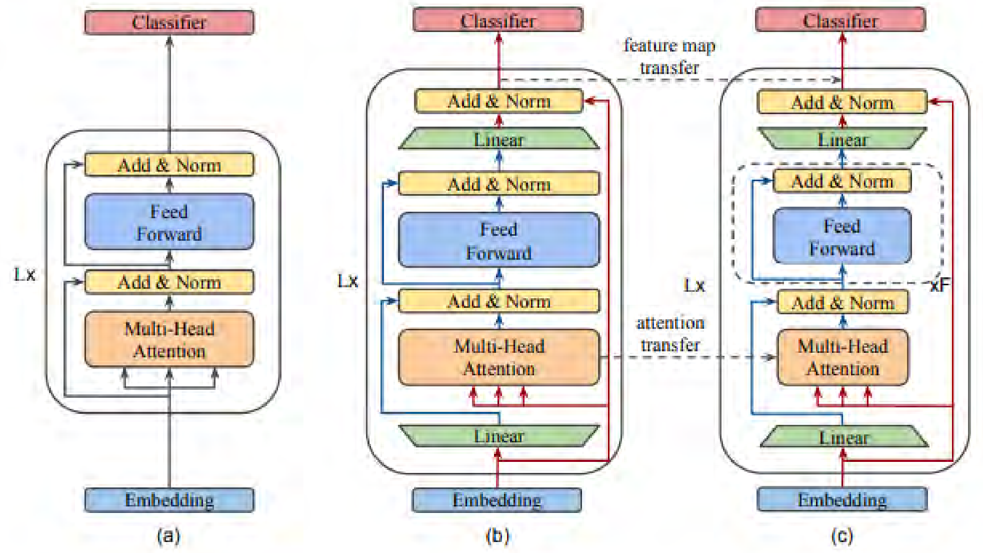


Figure 3.5: MobileBERT Architecture [19]

3.5.1 Architecture of MobileBERT

The architecture of MobileBERT is especially designed for devices which have limited resources like mobile phones. This compressed version of the BERT model has approximately 25M parameters.The inverted-bottleneck structures of MobileBERT enhance the feature map size and also ensure the information is correctly preserved [19]. This model has a similar number of layers but the layers are narrower than BERT and the depth and width are also balanced for optimal language processing. Additionally, the embedding layer usually large in models like BERT is compacted using dimensionality reduction techniques, adding to MobileBERT's reduced overall size while maintaining robust NLP capabilities.

3.6 DistilBERT

DistilBERT is a transformer-based model and a lighter version of BERT [18]. DistilBERT dispensing Teacher-Student distillation architecture allows it to reduce the size of a BERT model by 40%, while retaining 97% of its lingual comprehension capabilities and being 60% faster than BERT. This attenuated model having half (6 layers) of the layers of BERT (12 layers) and nominal 66 million parameters, not only outperforms BERT but also outsails GPT2 with 1.5 billion parameters and RoBERTa with 355 million parameters. Reduced parameters and eliminated layers by performing distillation architecture led this model to converge impetuously.The fig. 3.6 shows the architecture of DistilBERT model.

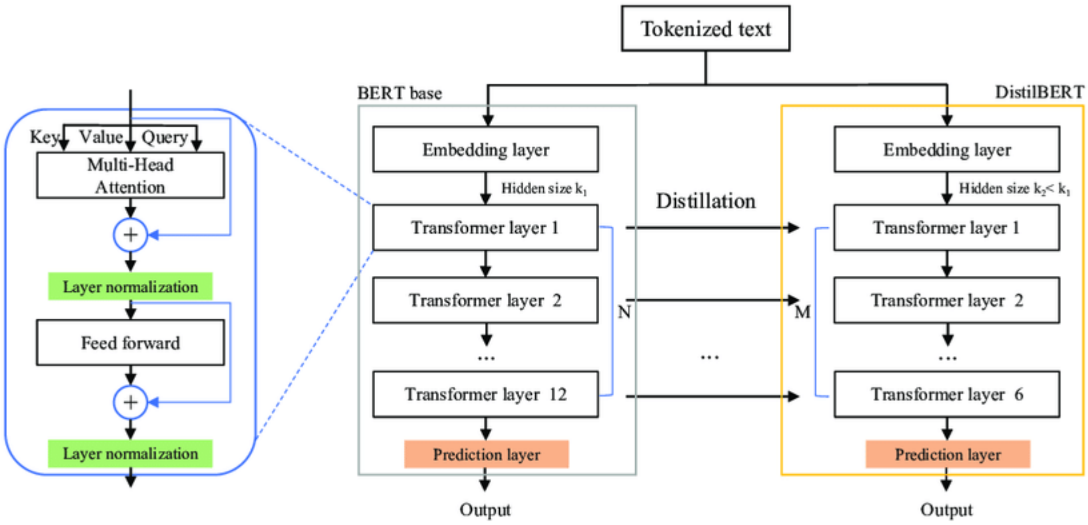


Figure 3.6: DistilBERT Architecture [24]

3.6.1 Architecture of DistilBERT

DistilBERT utilizes Teacher-student learning architecture. Where the teacher implies the conventional BERT model with frozen weights and as a predefined model. Whereas, the student implies the DistilBERT model which is privileged with the predefined weights and comprehension from the teacher (BERT) and converges faster utilizing the loss functions (misapprehension and learning) [18]. However, the distillation loss on the soft targets of the teacher and student model implies that the largest prediction coming from the teacher model is passed onto a softmax function with temperature and that provides with y. In the same manner, the rest of the predictions from the student model emerge y. This distillation loss assists calculation of entropy loss. Nevertheless, since the pooler layer and token-type embeddings along with 5 other layers (6 in total) are eliminated from BERT, therefore, Masked Language Modeling (MLM) loss is computed [18]. Here, input sentences are given to the model, and a few tokens are masked or hidden and the model is expected to output the complete original sentences. Thereby, it learns to guess the masked tokens. Eventually, end up comprehending the semantics of all the tokens that occur in the sentences. Thereafter, the cosine similarity of the hidden layers of buildings between the teacher and the student is compared. Mostly Teacher dissipation loss and teacher cosine similarity transfer knowledge to the student model weighted average of all these losses that are considered for finding the final loss, which is then used to back-propagate the student model.

3.7 ELECTRA

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) is a model like BERT but with a new pre-training approach which was developed in order to provide a faster learning thus making it more computationally efficient unlike BERT [15]. This alternative transformer model was introduced not only for the mentioned reason but also for considering the amount of cost and accessibility. The model works by detecting the corrupt tokens that generally arise in BERT as these tokens are one of the reasons for the longer computation cost. Moreover, the model outperforms BERT and RoBERTa. These large language models are learned towards contextual representations regardless of having the same data, size and compute(but uses $\frac{1}{4}$ of their pre-training computes). The approach trains two neural networks; one is a generator, G, and the other one is a discriminator, D. The fig. 3.7 shows the architecture of ELECTRA model.

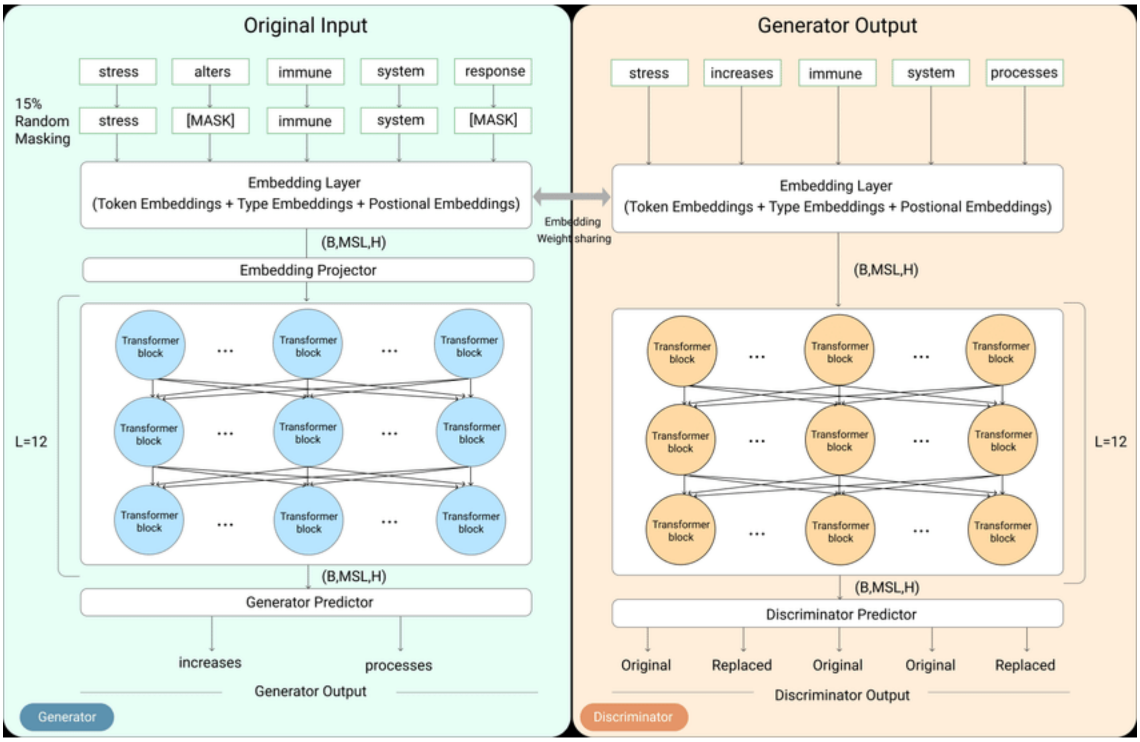


Figure 3.7: ELECTRA Architecture [21]

3.7.1 Architecture of ELECTRA

The architecture of the ELECTRA model is the discriminator one and the generator can be any other model, usually a small masked language model, and both of the models consist of an encoder [15]. In the generator part, the encoder maps the input tokens into a sequence of contextualized vector representations. During these mapping steps, few tokens get corrupted by the [MASK] token as the generator performs masked language modeling technique. The generator randomly selects some masking position to mask the input tokens. Afterwards, the mapped vectors are passed over for further training where the generator learns to predict the original position and the identity of the masked tokens. The discriminator is trained to distinguish tokens in the data from tokens that have been replaced by generator samples[15]. At first the discriminator is responsible for creating a corrupted, which solves the mismatched outputs of BERT, example by replacing the masked-out tokens with the generator's sample and then trains the discriminator to predict from each token whether it is an original or a replacement. An advantage of this task is that it learns from all the input tokens instead of the small masked subset of it. The discriminator uses the sigmoid function to determine if a token is original or a replacement, as shown in Equation 3.2.

$$D(x,t) = \text{sigmoid}(w^T h_D(x)t) \tag{3.2}$$

To deduce the following description of the models, the research meticulously assesses the effectiveness of these transformer-based models using CosmosQA Dataset. The main aim is to identify the model that outperforms others in the domain of commonsense based reasoning challenges, potentially advancing the field of Machine Reading Comprehension. These discoveries provide further insight into the substantial influence that transformer-based models can have on MRC, pushing the limits of natural language processing capabilities.

Chapter 4

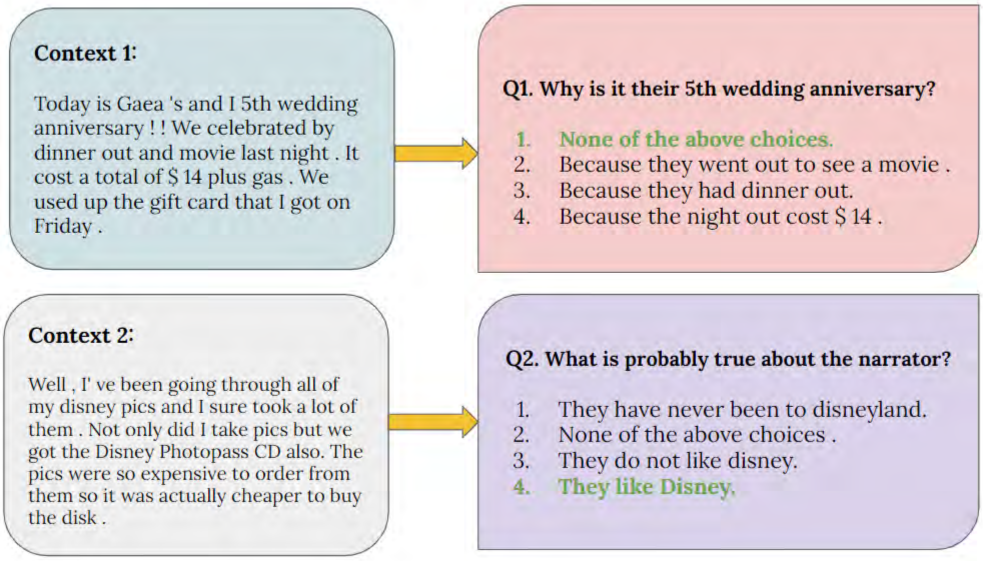
Description of the Data

4.1 CosmosQA

CosmosQA is one of the significant datasets which is specially designed for machine reading comprehension and commonsense reasoning in Natural Language Processing. The primary objective of this is to check NLP models or machines that can understand the information from the narrative texts as humans do. Additionally, this dataset generally focuses on the complexity and demand of commonsense reasoning [11]. The dataset is curated from various narratives from real life scenarios such as blogs and personal stories. Each context of the CosmosQA dataset is a narrative passage that provides the backdrop for the questions posed which range from straightforward to intricate by challenging the depth of comprehension of NLP models [11].

There are multiple choice questions in the dataset that require understanding beyond the text and tapping into the unspoken or implicit information. Unlike other datasets that focus solely on extractive question answering. CosmosQA dataset includes questions that are answerable only through inference and deduction exactly the way humans read between the lines. This dataset has three main components: a large number of narratives that comprise the instruction collection, a validation group for fine tuning the models and a test section designed to evaluate the generalization capabilities of NLP models to novel situations. So the evaluation of the CosmosQA dataset serves an important role in the advancement of Natural Language Processing(NLP) particularly those involving machine reading comprehension and complex question answering.The CosmosQA dataset was collected from the AI2 Leaderboard, which is a good resource for benchmarking and accessing various NLP datasets.The dataset files are in JSON format, which makes it straightforward to parse and preprocess the data for model training and evaluation.

As mentioned and explained previously on the paper, the dataset being used to carry out this research is CosmosQA. This is publicly accessible commonsense based question answering dataset which is being used to evaluate the performance of the transformer models. The figure **4.1** shows a example of the type of data present in the commonsense dataset, which includes a context, a question and four multiple-choice answers. The highlighted green lines represents the correct answer to the following questions.



Context 1:

Today is Gaea 's and I 5th wedding anniversary ! ! We celebrated by dinner out and movie last night . It cost a total of $ 14 plus gas . We used up the gift card that I got on Friday .

Q1. Why is it their 5th wedding anniversary?

1. None of the above choices.
2. Because they went out to see a movie .
3. Because they had dinner out.
4. Because the night out cost $ 14 .

Context 2:

Well , I' ve been going through all of my disney pics and I sure took a lot of them . Not only did I take pics but we got the Disney Photopass CD also. The pics were so expensive to order from them so it was actually cheaper to buy the disk .

Q2. What is probably true about the narrator?

1. They have never been to disneyland.
2. None of the above choices .
3. They do not like disney.
4. They like Disney.

Figure 4.1: Example of CosmosQA dataset.

4.2 Data Exploration

The CosmosQA dataset consists of 35,210 samples from a diverse range of questions and answers generated from thousands of passages taken from both real-world and fictional sources.It is divided into 25,262 training, 6963 testing and 2985 validation samples. It covers various topics, from science to literature and history. CosmosQA has multiple-choice questions that are answerable only through inference and deduction exactly the way humans read between the lines. In general, the answers demand reasoning and understanding mettle. For instance, questions like "What will happen if the incident x. . . ", "What might be the possible reasons for. . . " etc make a bid to comprehend the entire context line by line and respond similarly to the human brain processes.This dataset has three main components: a large number of narratives that comprise the instruction collection, a validation group for fine-tuning the models, and a test section designed to evaluate the generalization capabilities of NLP models to novel situations. Furthermore, all queries are tied with a paragraph(context), context ID that includes the options(probable correct answers-set), and the correct answer(label). Miscellaneous questions exist based on each context(paragraph/text). All contexts in the dataset have miscellaneous questions alike. Consequently, the CosmosQA dataset demonstrates superior efficiency in effectively managing complex inquiries. Hence, modeling the meaning of the offered text and situation is crucial. The fig. 4.2 shows the word cloud of CosmosQA dataset which indicates the frequency of different words.

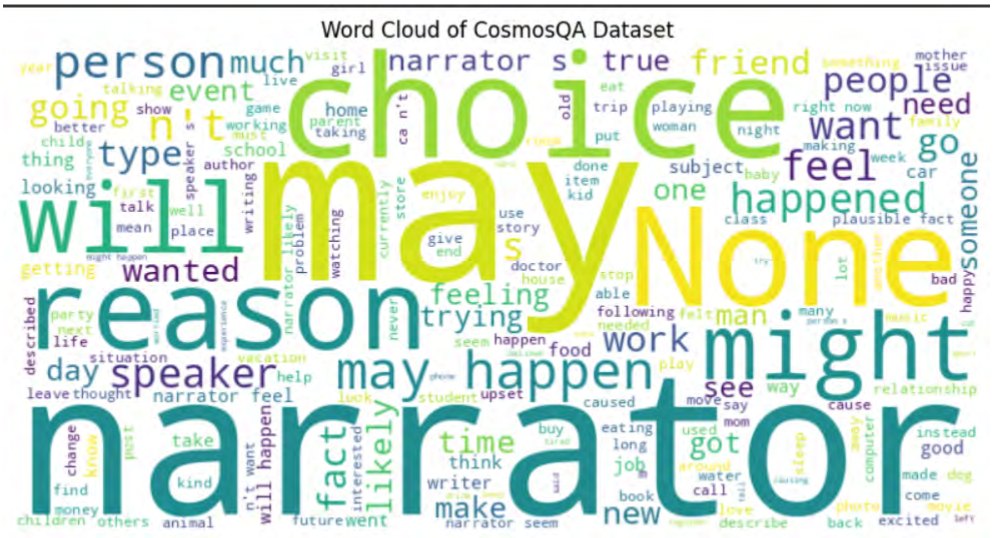


Figure 4.2: Word Cloud of CosmosQA Dataset

The distribution of context,question and answer lengths in the CosmosQA dataset are shown in fig. 4.3 ,fig. 4.4 and fig.4.5 . From the fig.4.3,we can see that the modal range for context paragraph lengths is approximately between 40 to 60 words. This is evidenced by the peak frequency within this interval, as indicated by the histogram's tallest bars. Moreover, the distribution is right-skewed which means there are more shorter paragraphs than longer ones and the frequency gradually decreases as the length increases.

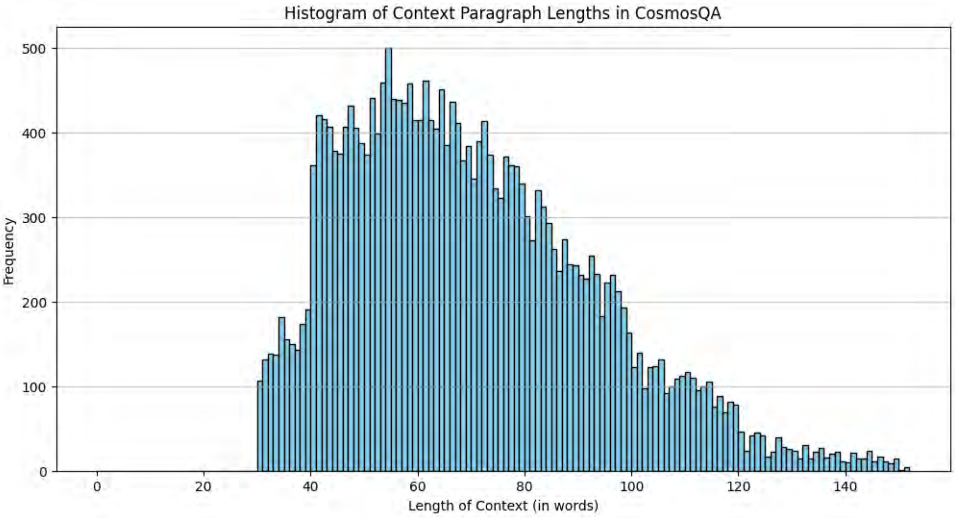


Figure 4.3: Context Length

The fig.4.4 shows the distribution of question lengths which is measured in the number of words.The modal range for question length is approximately 8 to 12 words, with the highest peak at around 10 words.The distribution is right-skewed which means shorter questions are more common than longer ones.The frequency gradually decreases as the question length increases. The questions longer than 20 words are rare in this dataset which indicates that most questions are concise and focused.

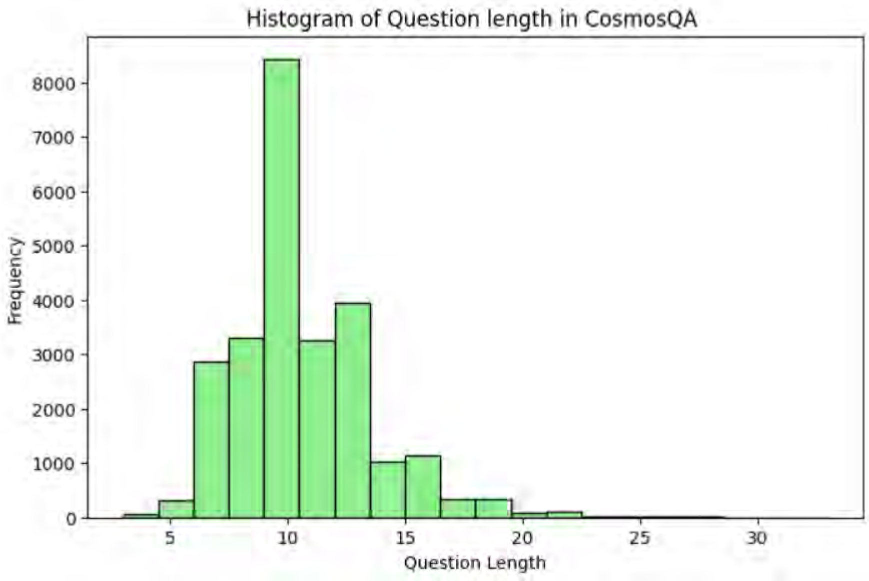


Figure 4.4: Question Length

On the other hand, fig. 4.5 indicates that short answers are very common in the CosmosQA dataset, especially answers that are only 1 or 2 words long. As the answers get longer, they become much less common. After about 10 words, it's rare to see answers that are longer and there are hardly any answers longer than 20 words.

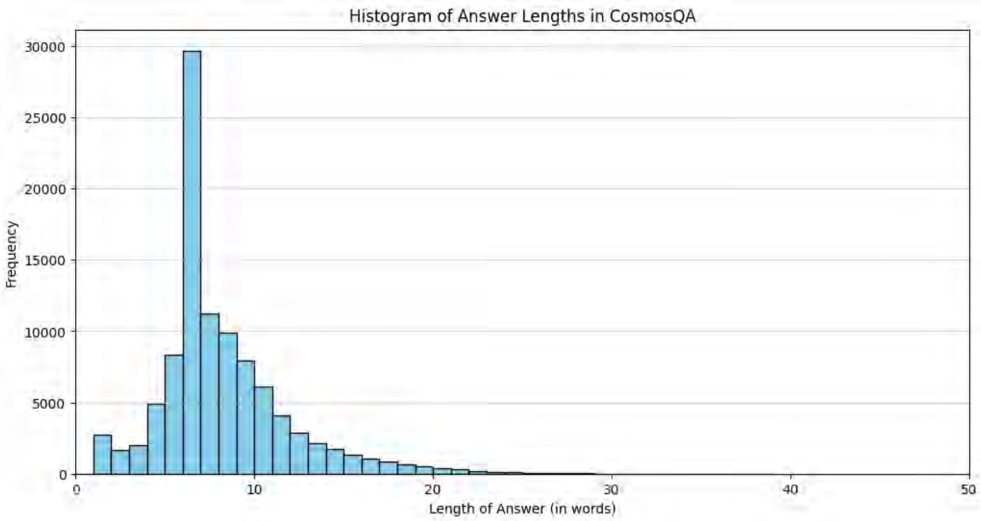


Figure 4.5: Answer Length

Chapter 5

Methodology

5.1 Working Plan

In order to carry out a comparative study of these algorithms, a good structured plan is necessary. Every step in the work plan should be done very carefully for extracting the most accurate results. Thus, here is a brief overview of the processing.

One of the first steps that needs to be fulfilled is to understand the problem of the given task. This step can be done by looking at the requirements for solving this issue. Afterwards, we need to collect a dataset and start with the data preprocessing. In this research, the primary benchmark dataset utilized is CosmosQA[11].The preprocessing which is conducted on this dataset includes tokenizing and organizing the data in a way which is suitable for the transformer based models to carry out the necessary tasks. In the next step, the preprocessed data is fed to the transformer models (BERT, ALBERT, RoBERTa, DistilBERT, MobileBERT and ELECTRA) to check their performance and to select the best performing transformer among them. The data is prepared for the selected model by converting the text into embeddings. The model is then trained on the prepared dataset by initializing the model with fine-tuning. The next step would be to evaluate performance of the model on the QA tasks. This procedure is performed by calculating the accuracy of the models. The transformer model is evaluated on the CosmosQA[11] to enhance its efficiency. Lastly, the model is optimized to reduce the size and data inefficiencies are also checked. Furthermore, regular monitoring and maintenance of the deployed model is also required to keep it updated on new corpuses, to ensure the model does not degrade in performance.

To summarize all the procedures, a flow-chart representation of the working plan is provided in fig. 5.1

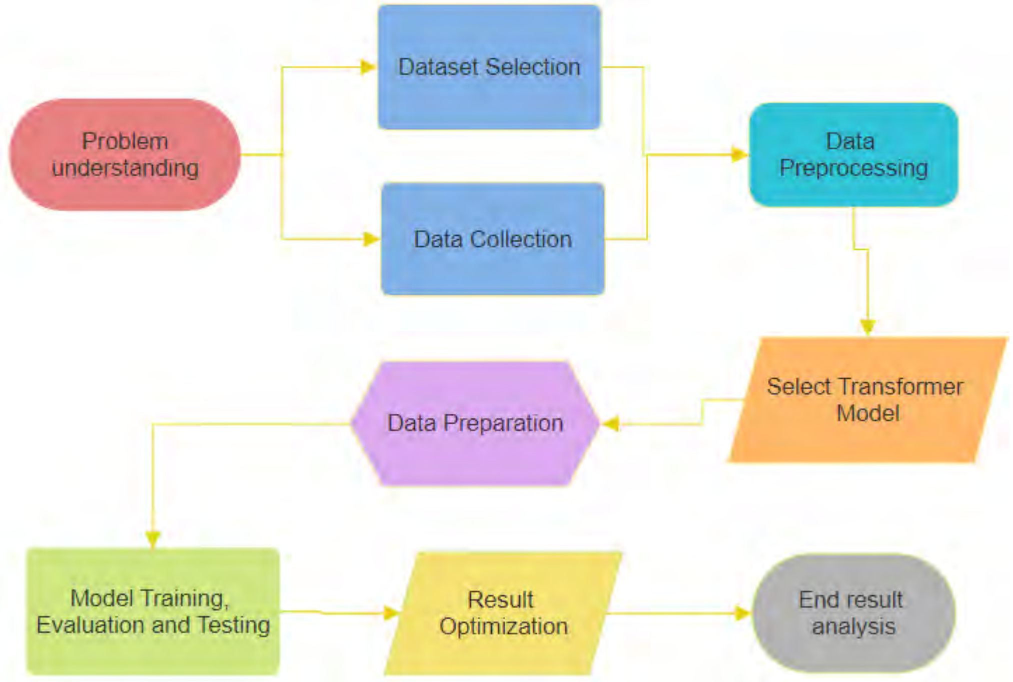


Figure 5.1: Work plan flowchart

5.2 Data Preprocessing

The preprocessing of the dataset consisted of several layers of task, in order to show an improvement in our work. The dataset, as previously mentioned, is a commonsense reasoning dataset and thus to achieve good performance from it a separate class was created for the model. However, before delving into the model file, the main preprocessing steps need to be discussed.

At first, the separate datasets for training, validation and testing were created by unravelling the data from the dataset folder which were shuffled beforehand and then unzipped. Afterwards, the input sequences were created and special tokens were added in front of each type of sequence: context, question and answer. In addition to that, two different lists were created, one for storing the contents of the datasets and the other one assigned for the labels. The labels section of the dataset contain all the options for the accurate answers. And hence while storing the information of the label dataset in the labels list, one-hot encoding method was used. Over here, if the correct answer for the input sequence is mentioned in labels, that position of the correct answer is denoted as a 1 and the rest positions as a 0, and this is also encapsulated in a list. The size of the list or the number of positions depends on the number of choices present in the dataset. A binary approach has been used in the model class, as it can facilitate the improvement of the model's performance and helps to bring flexibility in understanding contextual part of the QA dataset. It further helps reduce the dependency of multiple candidate answers. Furthermore, after the created datasets, the dataset loaders were also created. These data loaders are then passed over to the train and evaluation function where the training takes place. For the training process of the dataset, the model uses the content and label class which is passed into the model as batch.

5.3 The Binary Approach within a Multi-Class Framework

The `Model` class is constructed to perform sequence classification with a focus on selecting the correct answer from multiple choices provided for each question. It employs a transformer model, `AutoModelForSequenceClassification`, which is pretrained on a large corpus and fine-tuned here to classify individual answer choices as correct or incorrect. The model's architecture and forward propagation logic are specifically designed to handle the inputs structured for multiple-choice formats, making it highly suited for datasets such as CosmosQA.

Traditional multi-class classification directly computes a single output vector where each entry corresponds to the probability of each class. The class with the highest probability is selected as the prediction. In contrast, a binary approach within a multi-class framework treats each class decision as a separate binary classification task. Here, the model predicts whether each answer in contexts like question answering is correct or not, independently of the others.

In the traditional approach,the entire input sequence is formatted as `[CLS] Context [SEP] Question [SEP] Answer A [SEP] Answer B [SEP]` and so on, which is passed through the transformer models and a softmax layer at the end provide output probabilities for each answer.

In the binary approach,each answer is considered independently in a binary manner (correct or not).

- `[CLS] Context [SEP] Question [SEP] Answer A [SEP]`
- `[CLS] Context [SEP] Question [SEP] Answer B [SEP]`
- `[CLS] Context [SEP] Question [SEP] Answer C [SEP]`
- `[CLS] Context [SEP] Question [SEP] Answer D [SEP]`

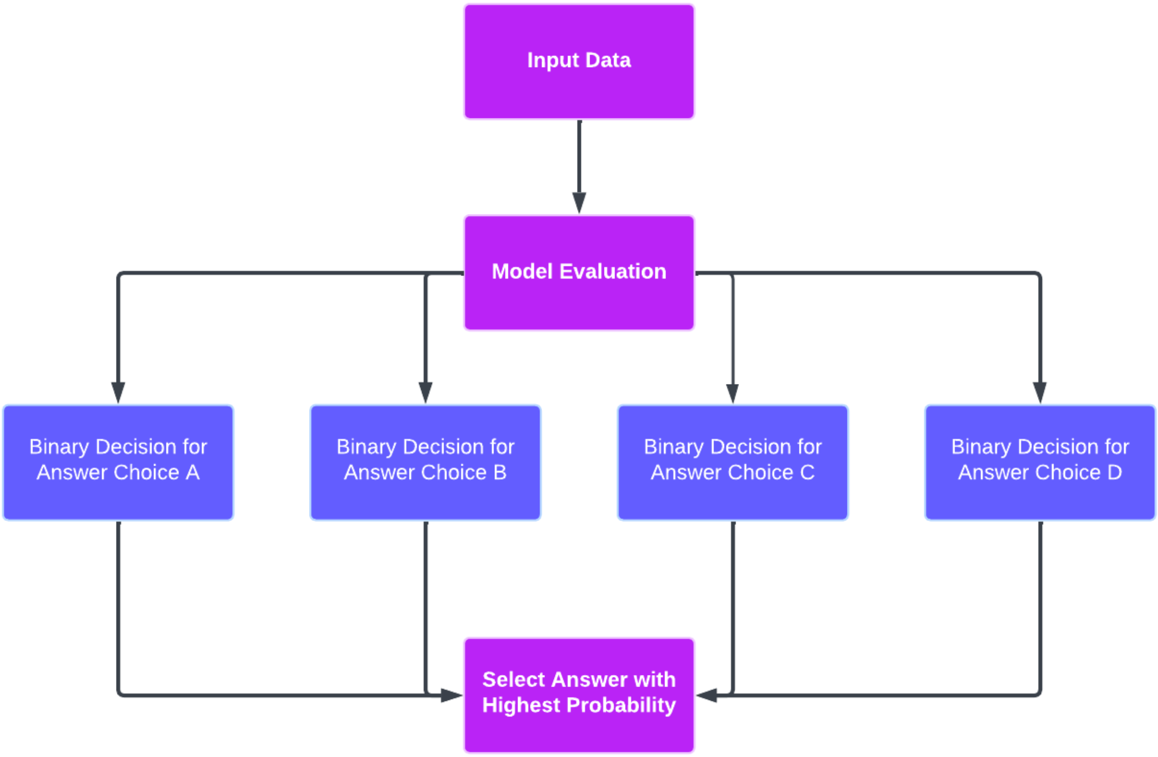

Input Data
Model Evaluation
Binary Decision for Answer Choice A
Binary Decision for Answer Choice B
Binary Decision for Answer Choice C
Binary Decision for Answer Choice D
Select Answer with Highest Probability

Figure 5.2: Flowchart of the Binary Decision Approach

The fig.5.2 shows how each sequences is processed through the model to predict a binary outcome (correct or incorrect). A binary classifier at the end of each sequence determines the likelihood of each answer being correct. The final decision is made by comparing these probabilities and selecting the answer with the highest probability as the correct one.

5.4 Hyperparameters

During the training of the transformer models multiple hyperparameters were implemented. The models were loaded from the hugging face transformer library and pytorch was used to import tools to facilitate the training of the models. To optimize the training of our model we have used the AdamW [3] optimizer for all of the models, and have used a learning rate of 3e-6 to 5e-6 in order to determine the outcome of our models. Moreover, the number of epochs assigned for training the dataset was 3 and a batch size 8 was used and lastly the maximum sequence length was kept at 512.

Chapter 6

Results and Analysis

For model evaluation purposes, the metric we are using for our comparative analysis of the models is the accuracy score. In this case, we are comparing the results achieved from the two accuracy scores, validation and test. The accuracy score is a metric which determines how frequently a machine learning algorithm can accurately predict the desired output. The intention behind using only accuracy score is to rationalize the performance of the models because the dataset class is imbalanced and hence the F1 score tends to be quite low compared to the accuracy score in this regard. The main paper of focus, which is the CosmosQA, carried out their performance evaluation on the basis of the accuracy score. The Equation 6.1 shows the mathematical representation of accuracy score.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

where:

- TP (True Positives) are the number of correct predictions that an instance is positive,
- TN (True Negatives) are the number of correct predictions that an instance is negative,
- FP (False Positives) are the number of incorrect predictions that an instance is positive,
- FN (False Negatives) are the number of incorrect predictions that an instance is negative.

6.1 Model Evaluation

The Table 6.1 shows the result from six transformer models-BERT, ALBERT, DistilBERT, RoBERTa, MobileBERT and ELECTRA; to evaluate the performance on commonsense reasoning using the CosmosQA dataset. Glancing at the outputs achieved in Table 6.1 the result indicates that even though the accuracy score from the validation dataset is high for both the RoBERTa and the ELECTRA model, being above 80%, the test dataset score is quite low, both around 65%. The model BERT and DistilBERT have slightly less inconsistency but are still quite noticeable, however ALBERT and MobileBERT do not have much difference and are consistent with good results of around 75% from both the models. The Table 6.1 shows the evaluation output we achieved so far.

Table 6.1: Results on CosmosQA dataset

Models	**Accuracy(Validation)**	**Accuracy(Test)**
BERT-base	76.60%	66.05%
ALBERT-base	74.98%	74.92%
DistilBERT-base	76.52%	67.64%
RoBERTa-base	80.13%	65.02%
MobileBERT	**75.00%**	**75.00%**
ELECTRA-base	81.86%	63.63%

6.1.1 Comparative Analysis of the Models

In order to make a comparative analysis of the models, the results from the Table 6.1 can be addressed. The differences between the accuracy scores indicates that overall the model is not performing that well as expected. One of the reasons for these discrepancies to occur is due to the architecture of the models. The RoBERTa model is an optimized version of the BERT model which was trained across a larger dataset than BERT and thus the model is highly dependent on the training data which may lead to the model to overfit on the validation dataset. On the other hand, the ELECTRA model due to its discriminator-generator architecture, the model might become sensitive to the characteristics of the validation dataset thus leading to a poor test dataset result. Moreover, due to the models being large with more than 100 million parameters and the dataset being small, one another reason would be that overfitting occurred on the validation dataset which hence lead to poor generalization of the test dataset causing a poor performance score.

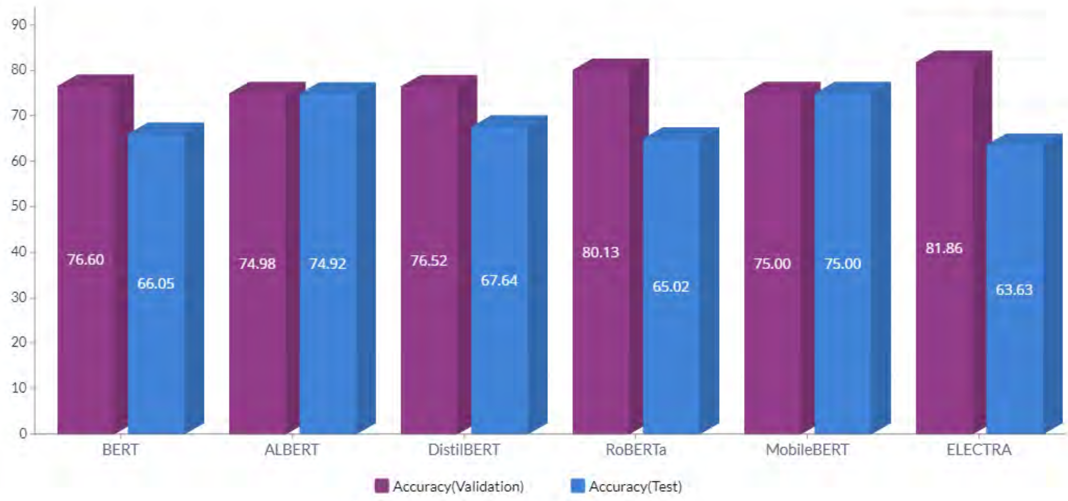


Figure 6.1: Graphical Representation of Model Performance

Analysis between models, BERT and DistilBERT can also be deduced from the Table 6.1 and the graph 6.1 as the two models showed a quite similar result on the CosmosQA dataset[11].The model BERT is a large and complex model hence it may not generalize tasks well on nuanced understanding. However, the accuracy score for the validation dataset using binary decision making approach which we used for the CosmosQA dataset gives a better score of around 76.60% whereas the result achieved using the multiway attention mechanism in the original paper for the dataset is 68.3%. Eventhough the result achieved from the validation dataset, is good which indicates the model is performing well but a drop in the test result, to 66.05%, shows that the overall performance of the model is not that good. The reason for this is because the model is overfitting on the validation dataset. Furthermore, the difference between the result achieved for the test dataset is also around 2%, the result even though is slightly higher in the original paper at around, 68.4%, using multiway attention but the binary decision making approach still performs well since the result is close of around 66.05%. Here, the slight difference in the scores might arise due to the hyperparameters used. As the hyperparameters used by the original paper are unknown it makes it harder to determine where the changes need to be made for a much better output and was well-suited. Additionally, the DistilBERT model also performed well on the validation set however the performance dropped once it evaluated the test set, even so the accuracy score on the test result is on par to the best result obtained from the original paper with around 68%. Even still, the issue with discrepancies relies strongly on the architecture of the model as this led to the overfitting issue on the validation dataset and the hyperparameters used. Moreover, DistilBERT tends to capture less complexity patterns as this variant of BERT has fewer layers which might have also resulted this model to perform moderately on the test set compared to models like ALBERT and MobileBERT.

Finally an analysis of the models, ALBERT and MobileBERT can be made. Both the models gave the best performance compared to the other models on this commonsense dataset. ALBERT gave 74.98% on the validation set whereas MobileBERT scored 75%. Moreover, the accuracy score on the test set was also 74.92% for AL-

BERT and MobileBERT outputs a result of 75%, like the validation set, which also indicates that no overfitting occurred for these two models. Moreover, among the two models, despite the results being very close MobileBERT slightly outperforms ALBERT by 0.008%. The scores achieved from these two models shows that the binary decision technique is a better approach in terms of dealing with and optimizing results from lightweight models which use smaller numbers of parameters more efficiently. The outcome obtained from these two models outperforms the highest result achieved on the original paper of the CosmosQA dataset. The transformer models ALBERT and MobileBERT contributed to a better result, of almost around 7% more, using binary decision approach than what BERT achieved following the multiway attention mechanism. Now, to give an overview on multiway attention, this is a technique which is a more extended approach to the traditional attention mechanism of the transformer models. This mechanism tends to handle multiple instances of input sequence and vectors at the same time. Whereas the binary approach separately handles the contents of the dataset for each multiple choice answers to understand the context better and gives a prediction based on it.

Now the reason why ALBERT and MobileBERT outperforms the rest is due to several factors, including how suitable the model is with the complexity and understanding of the dataset. Not all models would adapt well to different datasets, the interpretability depends on the models architecture and how well it generalizes the contents of the dataset. In this case, ALBERT yields a higher accuracy score because the model shares parameters across the layers in the model and hence even with fewer parameters it efficiently helps to generalize the contextual information better. To add on, the model uses SOP(Sentence Order Prediction) which pushes the model to understand more complex datasets sentence wise, which is very suitable for a commonsense structured dataset as this can facilitate the performance of such models by generating logical statements that are required to tackle the MCQ based dataset. SOP enhances the models ability to understand contextual relationships and can suggest inferences so that it can recognize better sentence patterns. MobileBERT, on the other hand, is a lightweight model which is suitable for capturing sequences of small datasets such as CosmosQA. This can lead to high performance efficiency due to smaller, faster and more agile architecture. It embeds the attributes of larger models such as BERT while giving a more optimized result in training due to proper usage of smaller parameters. Thus, these are some of the reasons why these two models, particularly, have outperformed the other models.

Chapter 7

Conclusion

Here, we have tried to address the challenges in understanding and answering complex textual information, commonsense and multi-reasoning questions by machine reading comprehension (MRC) systems. In order to improve the performance of MRC algorithms on CosmosQA dataset, we have explored the use of transformer-based models, including BERT, ALBERT, RoBERTa, DistilBERT, MobileBERT and ELECTRA. We had the vision of finding the perfect model that provides the best results for multi-reasoning and commonsense based tasks, comparing the performance of the models. The findings of this study can guide through the way of the development process of more effective, accurate, and efficient QA systems that can robustly handle complex questions and provide answers with closest accuracy. The study also provides an insight of how a binary decision approach in a multi-class framework gives a boost in the performance of the models in the commonsense field, where models typically tend to perform poorly. Furthermore, the challenging dataset showed a comparatively good result in lightweight models and the best performance result was attained from MobileBERT model with an accuracy of 75%. In conclusion, we can say that transformer-based algorithms have shown great potential in improving the performance of MRC systems and advancing the field of natural language processing.

For further research work, our main focus is to build a hybrid model out of the best two performing models in order to give a boost in the performance for this benchmark dataset. Therefore, we plan to achieve by storing the output values from one model and feeding the result across a different transformer model for a better F1 and accuracy score. The intention for our future work is to make the models more suitable in regards to tackling problem on commonsense based questions, as this type of questions can be very hard to understand by transformer models due to complexity in understanding the question pattern. Hybrid models thus in this case might facilitate the performance as these models tend to combine the best outputs leading to an improvement in the performance of the models. Additionally, some of the transformer models (BERT, RoBERTa, ELECTRA and DistilBERT) tend to overfit in our current approach, hence we would want to make sure to avoid this overfitting and the class imbalnce, which arises due to the miscalculation of the minory class, issues with a more refined technique during our future work.

Bibliography

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[2] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *International Conference on Learning Representations*, 2013.

[3] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, Dec. 2014.

[4] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. [Online]. Available: https://aclanthology.org/D14-1162.

[5] J. McAuley and A. Yang, “Addressing complex and subjective product-related queries with customer reviews,” in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW ’16, Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 625–635, ISBN: 9781450341431. DOI: 10.1145/2872427.2883044. [Online]. Available: https://doi.org/10.1145/2872427.2883044.

[6] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. [Online]. Available: https://aclanthology.org/D16-1264.

[7] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845a
Paper.pdf.

[8] A. Talmor, J. Herzig, N. Lourie, and J. Berant, “Commonsenseqa: A question answering challenge targeting commonsense knowledge,” *CoRR*, vol. abs/1811.00937, 2018. arXiv: 1811.00937. [Online]. Available: http://arxiv.org/abs/1811.00937.

[9] J. Welbl, P. Stenetorp, and S. Riedel, "Constructing datasets for multi-hop reading comprehension across documents," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 287–302, 2018. DOI: 10.1162/tacl_a_00021. [Online]. Available: https://aclanthology.org/Q18-1021.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[11] L. Huang, R. Le Bras, C. Bhagavatula, and Y. Choi, "Cosmos QA: Machine reading comprehension with contextual commonsense reasoning," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2391–2401. DOI: 10.18653/v1/D19-1243. [Online]. Available: https://aclanthology.org/D19-1243.

[12] J. Kiesel, M. Mestre, R. Shukla, *et al.*, "SemEval-2019 task 4: Hyperpartisan news detection," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 829–839. DOI: 10.18653/v1/S19-2145. [Online]. Available: https://aclanthology.org/S19-2145.

[13] Y. Liu, M. Ott, N. Goyal, *et al.*, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 `[cs.CL]`.

[14] H. Xu, B. Liu, L. Shu, and P. S. Yu, *Bert post-training for review reading comprehension and aspect-based sentiment analysis*, 2019. arXiv: 1904.02232 `[cs.CL]`.

[15] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, *Electra: Pre-training text encoders as discriminators rather than generators*, 2020. arXiv: 2003.10555 `[cs.CL]`.

[16] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *Albert: A lite bert for self-supervised learning of language representations*, 2020. arXiv: 1909.11942 `[cs.CL]`.

[17] J. Li, B. Wang, and H. Ding, "Lijunyi at semeval-2020 task 4: An albert model based maximum ensemble with different training sizes and depths for commonsense validation and explanation," Jan. 2020, pp. 556–561. DOI: 10.18653/v1/2020.semeval-1.69.

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*, 2020. arXiv: 1910.01108 `[cs.CL]`.

[19] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, *Mobilebert: A compact task-agnostic bert for resource-limited devices*, 2020. arXiv: 2004.02984 `[cs.CL]`.

[20] X. Chen, Z. Zhao, L. Chen, *et al.*, "WebSRC: A dataset for web-based structural reading comprehension," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4173–4185. DOI: 10.18653/v1/2021.emnlp-main.343. [Online]. Available: https://aclanthology.org/2021.emnlp-main.343.

[21] K. R. Kanakarajan, B. Kundumani, and M. Sankarasubbu, "Bioelectra:pretrained biomedical text encoder using discriminators," Jan. 2021, pp. 143–154. DOI: 10.18653/v1/2021.bionlp-1.16.

[22] U. Khanna and D. Mollá, "Transformer-based language models for factoid question answering at bioasq9b," *CoRR*, vol. abs/2109.07185, 2021. arXiv: 2109.07185. [Online]. Available: https://arxiv.org/abs/2109.07185.

[23] K. Pearce, T. Zhan, A. Komanduri, and J. Zhan, "A comparative study of transformer-based language models on extractive question answering," *CoRR*, vol. abs/2110.03142, 2021. arXiv: 2110.03142. [Online]. Available: https://arxiv.org/abs/2110.03142.

[24] H. Adel, A. Dahou, A. Mabrouk, *et al.*, "Improving crisis events detection using distilbert with hunger games search algorithm," *Mathematics*, vol. 10, p. 447, Jan. 2022. DOI: 10.3390/math10030447.

[25] X. Lv, Z. Liu, Y. Zhao, G. Xu, and X. You, "Hbert: A long text processing method based on bert and hierarchical attention mechanisms," *International Journal on Semantic Web and Information Systems*, vol. 19, pp. 1–14, Jan. 2023. DOI: 10.4018/IJSWIS.322769.

[26] T. Sultana, A. kumar mandal, H. Saha, M. Sultan, and M. D. Hossain, "Intent identification by semantically analyzing the search query," *Modelling*, vol. 5, pp. 292–314, Feb. 2024. DOI: 10.3390/modelling5010016.