# MACHINE READING COMPREHENSION USING DEEP LEARNING METHODS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE
OF
MASTER OF TECHNOLOGY
IN
**COMPUTER SCIENCE & ENGINEERING**

Submitted by:

**DEEPAK**
**2K20/CSE/08**

Under the supervision of
**Dr. Shailender Kumar**
(Professor)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

MAY, 2022

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

## CANDIDATE'S DECLARATION

I, Deepak, Roll No. 2K20/CSE/08 student of M. Tech (Computer Science and Engineering), hereby declare that the project Dissertation titled **"**Machine Reading Comprehension using Deep learning methods**"** which is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of and Degree, Diploma Associateship, Fellowship or other similar title or recognition.

| | |
|---|---|
| Place: Delhi | DEEPAK |
| Date: | 2K20/CSE/08 |

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

## CERTIFICATE

I hereby certify that the Project Dissertation titled **“Machine Reading Comprehension using Deep Learning methods”** which is submitted by Deepak, 2K20/CSE/08 Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

| | |
|---|---|
| Place: Delhi | Dr. Shailender Kumar |
| Date: | Professor |
| | Department of CSE |

# ACKNOWLEDGMENT

The success of this project requires the assistance and input of numerous people and the organization. I am grateful to everyone who helped in shaping the result of the project.

I express my sincere thanks to **Dr. Shailender Kumar**, my project guide, for providing me with the opportunity to undertake this project under his guidance. His constant support and encouragement have made me realize that it is the process of learning which weighs more than the end result. I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my work. They helped me throughout with new ideas, provided information necessary and pushed me to complete the work.

I also thank all my fellow students and my family for their continued support.

DEEPAK

2K20/CSE/08

# ABSTRACT

Machine Reading Comprehension (MRC) is a difficult Natural Language Processing (NLP) research subject with a broad range of practical applications. Its purpose is to create systems that can answer inquiries about a specific situation. The advent of large-scale datasets and deep learning has aided this field's rapid advancement in recent years. Despite the evident huge disparity between contemporary MRC models and true human-level reading comprehension, several MRC models have already outperformed human performance on numerous benchmark datasets.

“Multilingual Machine Comprehension" is a QA sub-task that comprises citing an answer to a question from a context, even if that answer written in a separate language from the excerpt itself. A lot of models have been trained to answer the question from a given short context which is a limitation of MRC, few models are considering this problem and adapting to handle the large input context to make the MRC more accessible and applicable to open domain scenarios. In this study, we examine Multilingual Representations for Indian Languages (MuRIL), rebalanced multilingual BERT (RemBERT), and XLM-RoBERTa, which are all BERT-based deep learning models. We trained these models to work on multilingual MRC particularly for two of the most used Indian languages Hindi and Tamil The datasets utilized in this study are freely available. The results of our research reveal that RemBERT outperformed other BERT-based deep learning models. For the dataset employed, the model received an F1 score of 84.58, an Exact Match of 74.05, and a Jaccard Index of 0.81.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

1. NLP: Natural Language Processing
2. IR: Information Retrieval
3. MRC: Machine Reading Comprehension
4. MLM: Masked Language Modeling
5. QA: Question Answering
6. TLM: Translation Language Modeling
7. NQ: Natural Questions
8. BERT: Bidirectional Encoder Representations from Transformers
9. XLM: Cross-lingual language model
10. EM: Exact Match

# CHAPTER 1

# INTRODUCTION

## 1.1 NATURAL LANGUAGE PROCESSING:

Humans can communicate with one another using natural language. They can communicate, exchange ideas, and grasp each other's perspectives, but the machine cannot. To function, the machine need instructions in a methodical manner. We must teach the computer to comprehend natural language so that people and robots can communicate. Natural language processing aids in the completion of these activities.

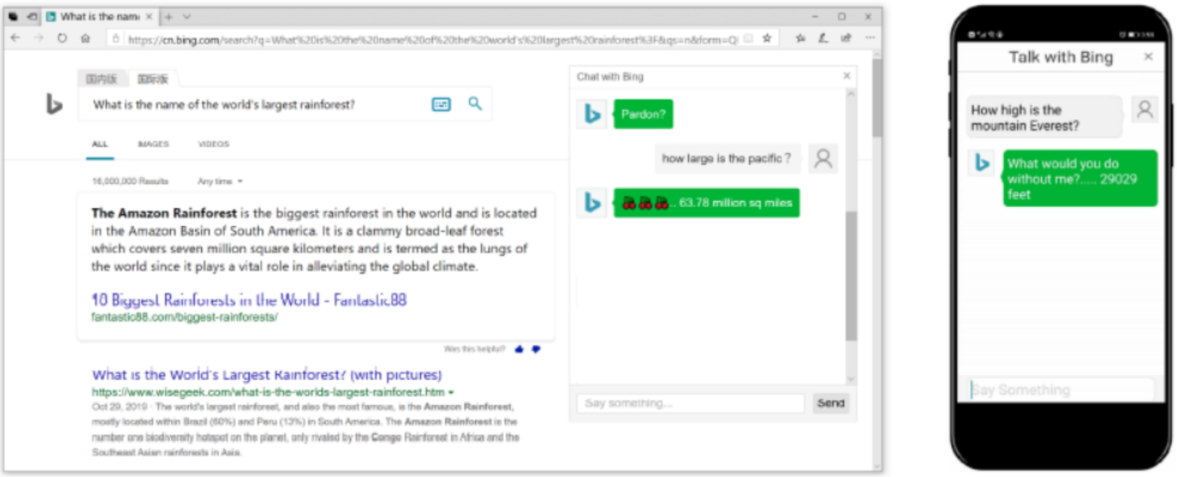
Natural Language Processing (NLP) is a major and newest field of computer science as well as artificial intelligence that allows people and computers to converse. Natural language and computer language interact in this process. It teaches the computer or gadget how to communicate with humans through voice or text. It enables machines to read, comprehend, and construct meaning from human language. Speech recognition, comprehension, and generation are all part of natural language processing.

## 1.2 QUESTION-ANSWERING:

Question-Answering (QA) is modeled as an Information Retrieval (IR) issue, in which structured databases, natural language publications, or web pages are used to automatically answer inquiries presented by people in natural language. Search engines return only the document or data relevant to a query, while QA systems provide useful information from those sites. Two important variables that impact the quality of a QA system's output are the range of information resources used to get the result and the interpretation of that information in order to effectively provide the required response. Even for a basic query, getting correct information in today's age of information explosion necessitates a lot of compute and evaluation resources. As a result, the majority of study in this field focuses on the second element. In contemporary NLP work, this feature of grasping the intrinsic contextual information is referred to as Machine Reading Comprehension (MRC).

Many Indian languages, which are difficult to type into our existing technologies, have expanded in use as a result of greater availability to mobile devices and speech recognition systems. There has been an increase in resources for languages like Hindi and Tamil, but user penetration has not kept pace. Developing systems that can answer common questions about the web is a critical first step in making it even more useful. There has been very little research in this area for Hindi, and Tamil, even fewer datasets are available to the general public as a result.

A lot of NLP systems, like a search engine and a discussion system, might benefit from machine reading comprehension. As demonstrated in Figure 1, currently, when we type a query inside the Bing search engine, it might occasionally provide the proper answer by highlighting it in the context. Furthermore, if we access the "Chat with Bing" section of the Bing website, as displayed on the right half of the internet browser shown in Figure 1.1, we can ask it questions like "How big is the Pacific?" and the chatbot provided by Bing will respond with "63.78 million square miles." We may also access this "Chat with Bing" on Bing's App, as seen in the right half of Figure 1.1. It is obvious that machine reading comprehension can assist enhance the speed of these search engines and conversation systems, allowing users to receive the proper response to their inquiries more quickly and reducing the strain of customer care representatives.



What is the name

https://cn.bing.com/search?q=What%20is%20the%20name%20of%20the%20world's%20largest%20rainforest%3F&qs=n&form=Q

国内版 国际版

What is the name of the world's largest rainforest?

ALL IMAGES VIDEOS

16,000,000 Results Any time

**The Amazon Rainforest** is the biggest rainforest in the world and is located in the Amazon Basin of South America. It is a clammy broad-leaf forest which covers seven million square kilometers and is termed as the lungs of the world since it plays a vital role in alleviating the global climate.

10 Biggest Rainforests in the World - Fantastic88
fantastic88.com/biggest-rainforests/

Was this helpful?

What is the World's Largest Rainforest? (with pictures)
https://www.wisegeek.com/what-is-the-worlds-largest-rainforest.htm
Oct 29, 2019 · The world's largest rainforest, and also the most famous, is the **Amazon Rainforest**, mostly located within Brazil (60%) and Peru (13%) in South America. The **Amazon Rainforest** is the number one biodiversity hotspot on the planet, only rivaled by the Congo Rainforest in Africa and the Southeast Asian rainforests in Asia.

Chat with Bing

Pardon?

how large is the pacific ?

... 63.78 million sq miles

Say something...

Send

Talk with Bing

How high is the mountain Everest?

What would you do without me?..... 29029 feet

Say Something

Figure 1.1. A Machine Reading Comprehension example in Bing search engine

Machine reading comprehension should be an automatic job that finds the answer to a question from a given context in Indian languages Various techniques such as CNN, RNN, LSTM, and BERT have been presented for Deep Learning and Natural Language Processing (NLP) models BERT based deep learning models have demonstrated to be effective in a a wide range of natural language processing activities like machine reading comprehension and text summarization according to studies The key types of Bert based deep learning models utilized for multilingual MRC are MURIL, RemBERT, and XLM-Roberta which we will discuss and compare in this study RemBERT Large outperformed the other models according to the results. Figure 1.2 [15] shows the common machine reading comprehension system where answer is predicted by feature extraction from question and context embeddings.

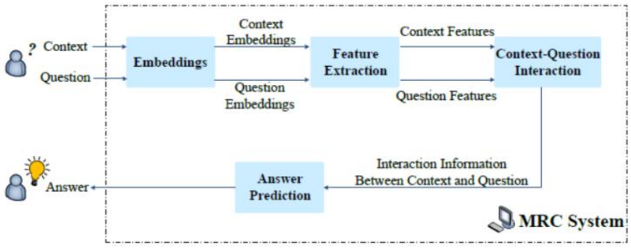
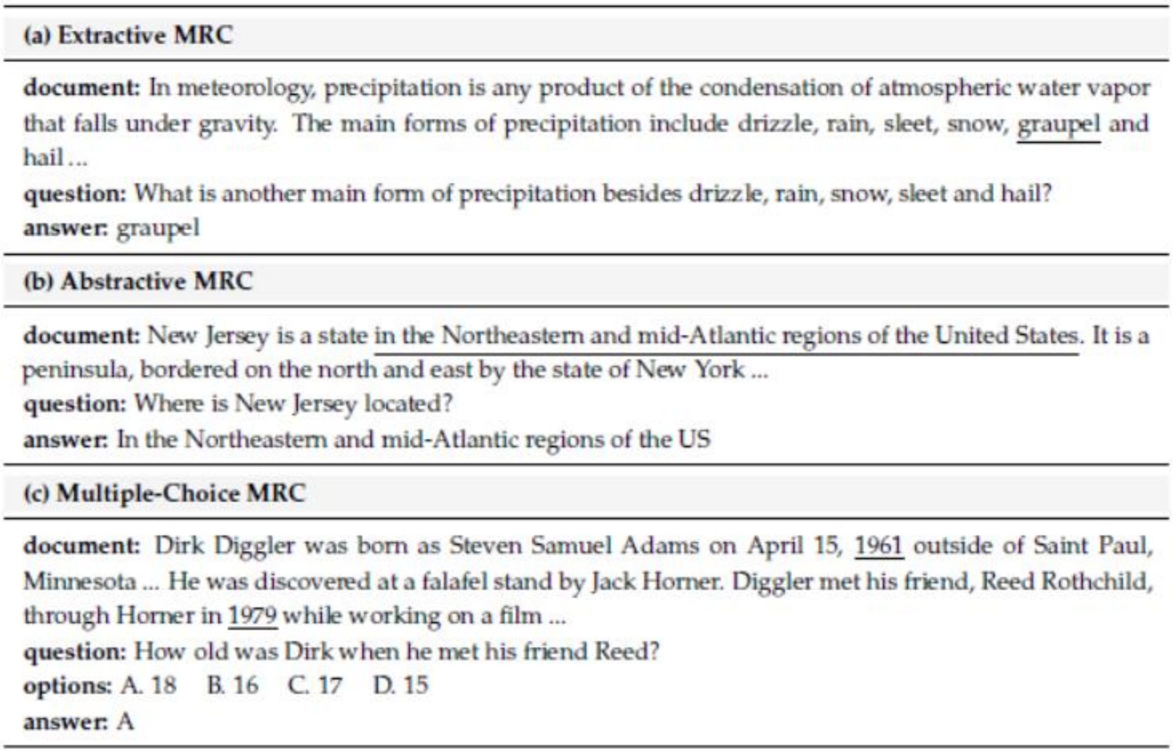


Figure 1.2. Machine Reading Comprehension system architecture

There are several MRC tasks and classification methods. We may categorise MRC tasks into three types according on the kind of answer: extractive, abstractive, and multiple-choice.

(a) Extractive MRC is a problem in which the solution is a small span of text that can be retrieved from a document. An example is shown in Table 1.1 (a). The fact that many replies cannot be expressed as a span of the text is a key drawback of extractive MRC.

(b) In response to the limitations of extractive MRC, abstractive MRC tasks use human-generated texts that are not required to represent document spans, as illustrated in Table 1.1. The assessment of abstractive MRC, on the other hand, is difficult because to the wide range of potential responses. Furthermore, since annotators often duplicate spans as replies, the bulk of solutions in many of these jobs are still extractive.



**(a) Extractive MRC**

**document:** In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail ...

**question:** What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

**answer:** graupel

**(b) Abstractive MRC**

**document:** New Jersey is a state in the Northeastern and mid-Atlantic regions of the United States. It is a peninsula, bordered on the north and east by the state of New York ...

**question:** Where is New Jersey located?

**answer:** In the Northeastern and mid-Atlantic regions of the US

**(c) Multiple-Choice MRC**

**document:** Dirk Diggler was born as Steven Samuel Adams on April 15, 1961 outside of Saint Paul, Minnesota ... He was discovered at a falafel stand by Jack Horner. Diggler met his friend, Reed Rothchild, through Horner in 1979 while working on a film ...

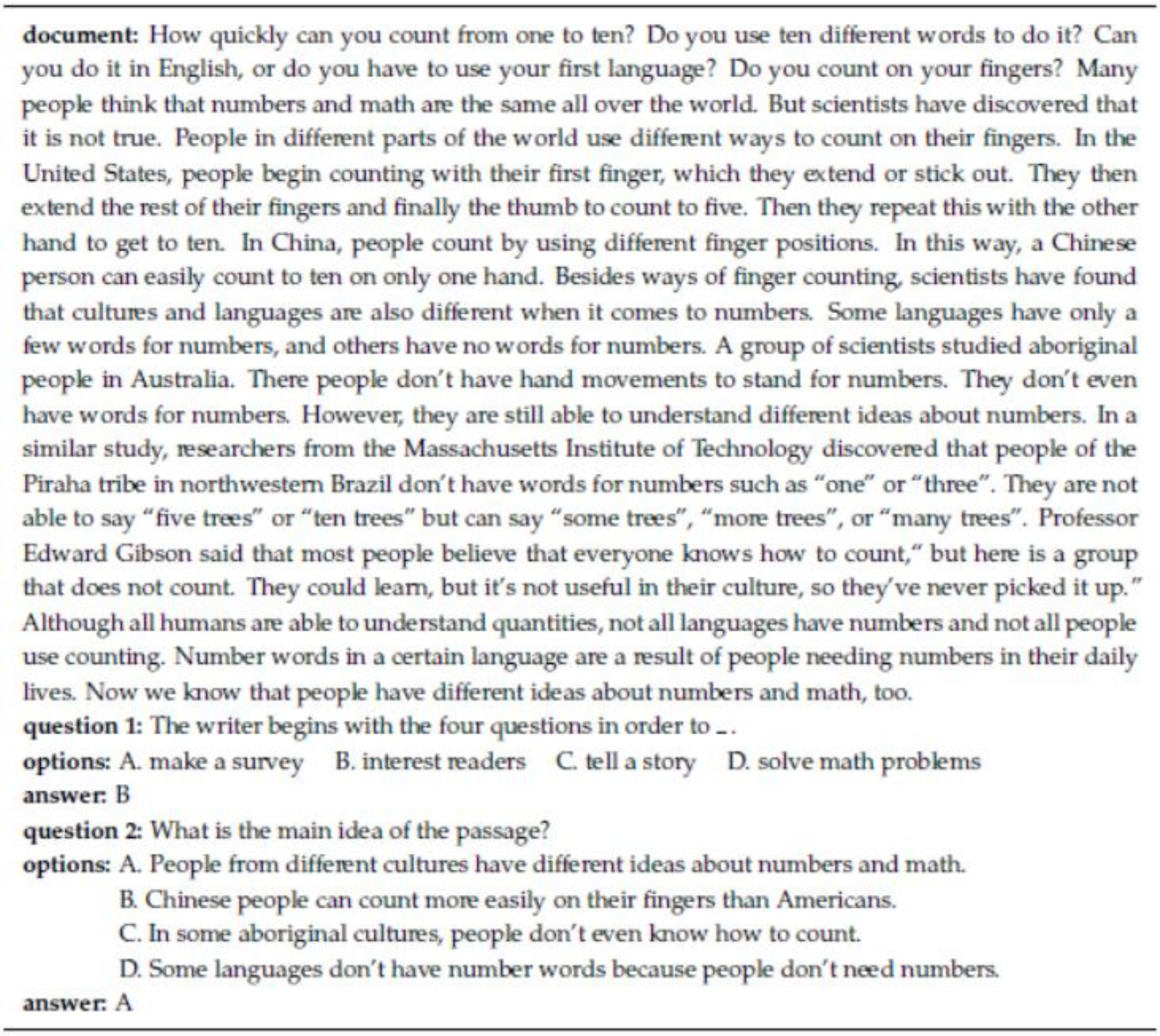
**question:** How old was Dirk when he met his friend Reed?

**options:** A. 18 B. 16 C. 17 D. 15

**answer:** A

Table 1.1. Examples adapted from representation MRC tasks: (a) SQuAD, (b) CoQA , 2019), and (c) MultiRC.

(c) In multiple-choice MRC, a question is accompanied with numerous response alternatives, and the aim is to choose the proper option (s). An example is shown in Table 1.1. In comparison to abstractive MRC, we can more readily use objective assessment criteria like accuracy to assess system performance.



**document:** How quickly can you count from one to ten? Do you use ten different words to do it? Can you do it in English, or do you have to use your first language? Do you count on your fingers? Many people think that numbers and math are the same all over the world. But scientists have discovered that it is not true. People in different parts of the world use different ways to count on their fingers. In the United States, people begin counting with their first finger, which they extend or stick out. They then extend the rest of their fingers and finally the thumb to count to five. Then they repeat this with the other hand to get to ten. In China, people count by using different finger positions. In this way, a Chinese person can easily count to ten on only one hand. Besides ways of finger counting, scientists have found that cultures and languages are also different when it comes to numbers. Some languages have only a few words for numbers, and others have no words for numbers. A group of scientists studied aboriginal people in Australia. There people don't have hand movements to stand for numbers. They don't even have words for numbers. However, they are still able to understand different ideas about numbers. In a similar study, researchers from the Massachusetts Institute of Technology discovered that people of the Piraha tribe in northwestern Brazil don't have words for numbers such as "one" or "three". They are not able to say "five trees" or "ten trees" but can say "some trees", "more trees", or "many trees". Professor Edward Gibson said that most people believe that everyone knows how to count," but here is a group that does not count. They could learn, but it's not useful in their culture, so they've never picked it up." Although all humans are able to understand quantities, not all languages have numbers and not all people use counting. Number words in a certain language are a result of people needing numbers in their daily lives. Now we know that people have different ideas about numbers and math, too.

**question 1:** The writer begins with the four questions in order to _.

**options:** A. make a survey B. interest readers C. tell a story D. solve math problems

**answer:** B

**question 2:** What is the main idea of the passage?

**options:** A. People from different cultures have different ideas about numbers and math.
B. Chinese people can count more easily on their fingers than Americans.
C. In some aboriginal cultures, people don't even know how to count.
D. Some languages don't have number words because people don't need numbers.

**answer:** A

Figure 1.3. Examples of various types of questions from a non-extractive MRC dataset RACE (Lai et al., 2017).

We concentrate on non-extractive multiple-choice MRC problems in which a large majority of the response alternatives are not extractive text spans. Apart than surface matching, there are many sorts of complex questions like as arithmetic word problems, summarization, logical reasoning, and sentiment analysis in comparison to questions in extractive MRC tasks. We concentrate on non-extractive multiple-choice MRC problems in this dissertation, in which a considerable fraction of response alternatives are not extractive text spans. Apart than surface matching, there are many sorts of hard questions in extractive MRC tasks, including as arithmetic word problems, summarization, logical reasoning, and sentiment analysis, which need sophisticated reading abilities and previous world knowledge. To answer the question in Table 1.1 (c), for example, you'll need arithmetic skills; to answer the questions in Figure 1.3, you'll need common sense and summarising skills.

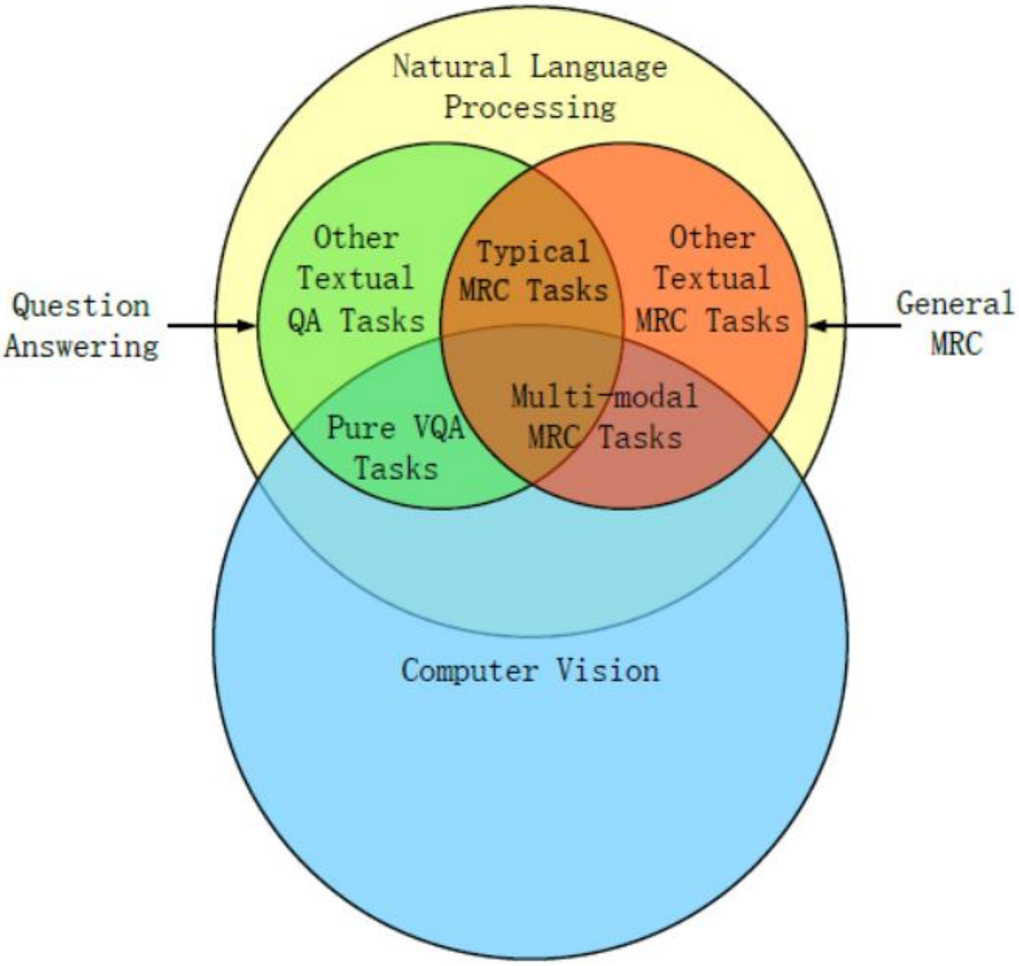


Figure 1.4. Question-Answering (QA), Machine reading comprehension (MRC), Computer versioning (CV), Natural language processing (NLP): Relation

## 1.3 MULTILINGUAL MACHINE READING COMPREHENSION (MMC):

A key challenge for Q-A systems is the ubiquity of knowledge bases in several languages, which restricts the origin of monolingual systems. This may be efficiently managed by multilingual systems that are able to understand the semantic, syntactic structure of many languages at the same time. Because they can comprehend queries in one language and reply using resources in another, Multilingual Question-Answering (MQA) systems are the name for these systems. MQA is a crucial need in IR, especially for non-Latin languages, since the bulk of material on the internet is written in English, making searches in languages such as Hindi, Mandarin, and Japanese very difficult. MMC (Multilingual Machine Reading Comprehension) is a subset of MQA in which the context is supplied as a text extract and the answer is sought as a text span. a single, deep contextualized language model m-BERT that is pre-trained from monolingual corpora available in 104 languages, is a recent multilingual variation of BERT, that has delivered cutting edge results across numerous language pairings including English-Japanese also English-French. Unexpectedly, m-BERT is also excellent at cross-lingual model transfer with zero-shot, which entails fine-tuning the model for evaluation in another language using task-specific annotations in one language. This inspire us to make use of m-BERT in English-Hindi Multilingual MRC contexts where it has not yet been MQA tested. The most current cutting edge uses a sequential strategy with integrated English and Hindi teaching to address the difficulty. One of the most significant drawbacks of English-Hindi Multilingual MRC research is the lack of a standardized evaluation dataset. The previous evaluated a sub-section that is translated from SQuAD dataset, but it's not available publicly also requires enormous pre-processing due to the data not being in SQuAD specific format and the answers which are translated having many inconsistencies due to machine translation. Despite the fact that we were able to get the authors' dataset and pre-process it to do comparison with the published results, For future studies, we do not advocate it. XQuAD recently introduced by DeepMind as a multilingual assessment benchmark, that contains subset data from SQuAD v1.1 translated into 10 languages contains Hindi also. We augment this dataset by making cross-lingual versions (for example, English Question, Hindi Answer pair or vice-versa) and publishing our results on the XQuAD dataset that we advocate as the new evaluation standard for further work in the field.

## 1.4 APPLICATIONS:

Question answering has been used in a variety of sectors of study. The three domains in where QA systems are employed are Closed-Domain, Open-Domain, Restrict-Domain, and. Our study described where and how researchers put their systems into practise. There were no applications in Closed-Domain. With this study, we discovered that Open-Domain implementations based on the WWW account for the majority of research, and medicine are treated at a high rate, as shown in Fig. 1.5. We can see how question-answering overlaps with large number of sectors and the domains in this research, demonstrating how these systems may be useful for information extraction for every kind of user and requirement.

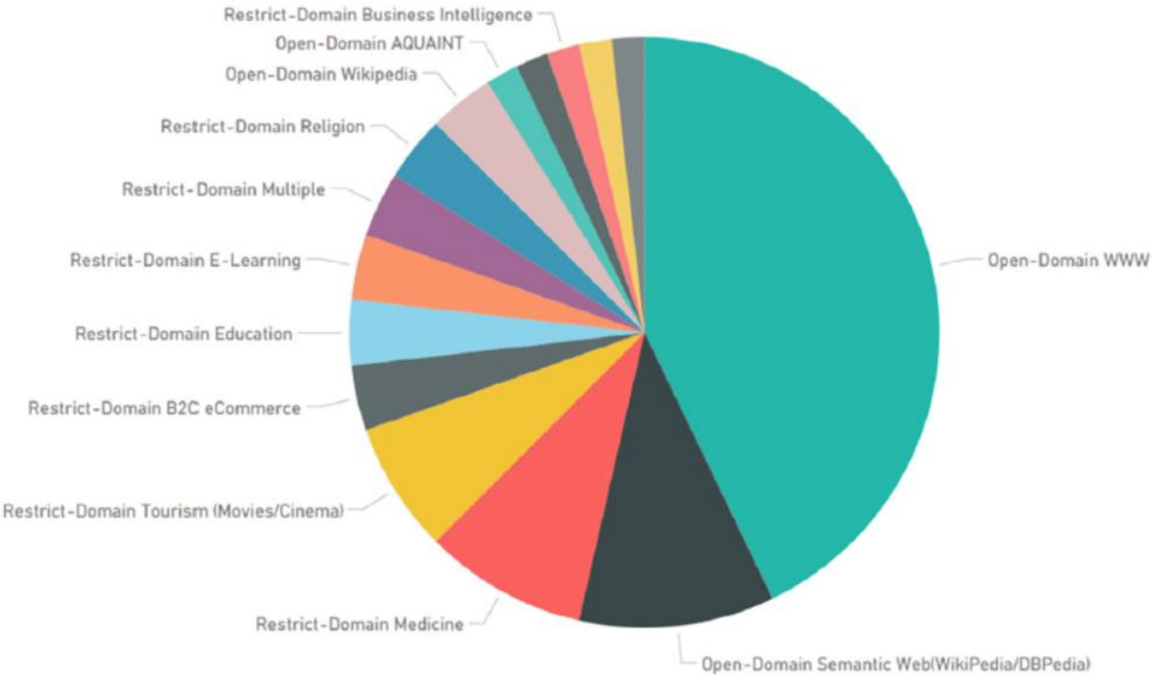


Figure 1.5. Fields where researchers are putting their theories into practice.

### 1.5 CHALLENGES:

The problems and probable future research trends are listed below. Reading comprehension that can be explained: The majority of models use naive similarity-based matching. They don't give a damn about gathering and analysing evidence, which leads to them failing to answer problems that demand more advanced thinking or being easily diverted by completely unrelated statements. Recent work has begun to concentrate on enhancing the explainability of models, such as the ability to reason across various cues. The most essential issue for machine comprehension researchers is and will continue to be improving computers' actual language understanding. The age of explainable machine reading comprehension is just getting started.

Managing vast amounts of data: Many models are bound by basic situations in which the solution may be found in a single paragraph. Few models take into account the difficulty of adapting the model to huge input, which is critical for industrial applications such as constructing an open-domain QA system. The challenge becomes much more difficult when complicated reasoning and huge input are combined. The crucial problem of entity representation is addressed by GCN-based multi-hop reasoning systems .Nonetheless, they depend on the scenario in which a candidate list is provided, implying that they do not address the challenge of evaluating a complicated and large entity relation network. Future research will focus on narrowing the potentially vast search space and carefully filtering out extraneous data. Furthermore, future datasets should place a greater emphasis on evaluating the capacity to deduce the answer from huge texts.

Users are frequently more concerned with the variety of inquiry kinds and formats. Typically, newly reported datasets put a lot of work into covering a variety of question kinds and areas. Still, as Yatskar points out, there is room for improvement in the future (2018). Furthermore, developing a consistent knowledge representation is critical for future work aimed at making the model less personalised. Developing an intelligent conversational system has already been a long-term objective for AI researchers (Gao et al., 2018), and researchers are currently focusing on adding dialogue into their models. Future research should consider how to overcome present restrictions and provide more appropriate datasets to aid the development of conversational machine comprehension QA systems.

# CHAPTER 2

# PRIOR WORK

## 2.1 HISTORY:

Researchers working on computer-implemented narrative comprehension models (Charniak, 1972) identified responding questions about paragraphs of text as a task criteria for measuring language understanding systems' reading abilities in the 1970s (Lehnert, 1977). In the 1980s and early 1990s, however, the field was mostly ignored (Chen, 2018). Hirschman et al. (1999) generated a dataset that sparked renewed interest. The dataset is tiny, and the systems are primarily dependent on rules. In the 2010s, researchers began to formulate MRC as a supervised learning, and a growing number of large-scale datasets were created, greatly facilitating the development of machine learning-based approaches. It's worth noting that question-answering (QA) machine reading comprehension (MRC) are closely connected, and researchers today often use the two words interchangeably rather than distinguishing between them. MRC tasks, have traditionally been text-dependent. they focus on testing machine readers' comprehension of a given text by asking a model to answer questions related to the text. Many question answering tasks do not provide a ground truth document with each question, making them less suitable for detecting MRC progress.

## 2.2 MRC Models:

Rule-based models, classical machine learning models, and end-to-end neural models are the three types of MRC models.

1) Rule Based Model. The majority of early MRC models are built on hand-crafted rules, with rule-based pattern matching (e.g., bag-of-words matching) and shadow linguistic processing (e.g., stemming) being the most common (Hirschman et al., 1999; Riloff and Thelen, 2000). For multiple-choice MRC, for example, standard

rule-based models calculate the matching score among each question-option combination and the reference document, then choose the option with the best score as the answer. Simple guidelines such as the count of matched words (Yih et al., 2013) and the sum of the TF-IDF values of the matched words are used to determine the matching score (Richardson et al., 2013).

2) Model of traditional machine learning. Machine learning-based techniques have grown in popularity during the 2010s, thanks to the availability of training data. Early machine learning techniques focused on extracting rich information and using traditional machine learning algorithms (Sachan et al., 2015;Wang et al., 2015). In a max-margin learning framework, Wang et al. (2015) construct a model for multiplechoice MRC that includes features based on rule-based pattern matching, dependency syntax, frame semantics, coreference, and word embeddings.

3) A complete neural model. Since 2015, when large-scale training data for MRC became accessible, machine learning-based techniques for MRC have been steadily moving in the direction of deep learning models (Hermann et al., 2015). Deep learning models depend on hand-crafted features much less than conventional machine learning models. Deep learning models, in contrast, use end-to-end neural networks to learn the features themselves. For multiple-choice MRC, for example, a typical deep learning model turns the document, question, and option into embedding vectors and delivers them to a neural network with various modelling or interaction layers. The neural network is trained to determine whether or not the choice is accurate (Wang et al., 2018d). Prior to 2018, most studies used a random initialization of neural network parameters before tuning them using the gradient descent approach or variations on the target MRC task's training data. Radford et al. (2018) suggest pre-training the neural network with a language model goal across large-scale corpora such as thousands of books, then fine-tuning the pre-trained neural network on the target MRC task. This framework has had a lot of success in MRC and is still used in today's cutting-edge MRC models.

An investigation into machine reading comprehension is now underway. Many researchers are working on automating MRC task in many languages. With the publication of a number of benchmark datasets, MRC research has sparked a lot of interest.

From pretrained word embeddings to pretrained contextualized representations [24] to transformer-based language models [23], unsupervised learning representation has considerably enhanced highly developed in natural language processing. Parallel research on cross-lingual understanding [12][21] broadens these systems to incorporate other languages and the cross-lingual environment, in which a model can be trained and worked in different languages.

XLM and m-BERT are masked language models that are trained on various languages without cross-lingual supervision. Translation language modelling (TLM) [12] is an approach for gaining the cross-lingual natural language inference which is state of the art (XNLI) [25] benchmark by using parallel data. In addition, they show substantial progress in unsupervised machine translation and sequence synthesis pretraining.

Multilingualism comes naturally in bottleneck designs since the monolingual BERT representations [24] are equivalent across languages. However, all of previous work was done on a far smaller scale in respect of training data than our method.

The benefits of increasing the model size and training data while scaling language model pretraining have been widely investigated in the literature. When trained on billions of tokens in a monolingual setting, LSTM models which are large scale may achieve much superior achievement on language modelling benchmarks [24]. GPT [13] emphasizes the importance of scaling the quantity of data, while RoBERTa [10] shows that training BERT on more data over a longer period of time improves performance significantly. We demonstrate that XLM and m-BERT are under tuned, and that small adjustments to the unsupervised MLM learning approach drastically improve performance.

We train using cleaned CommonCrawls [5], which quadruple the quantity for low-resource languages data on average. Identical data has been presented to be useful in collecting high-standard word embeddings in a variety of languages.

# CHAPTER 3

# PROPOSED WORK

## 3.1 PROBLEM STATEMENT

Many Indian languages, which are difficult to type into our existing technologies, have expanded in use as a result of greater availability to mobile devices and speech recognition systems. There has been an increase in resources for languages like Hindi and Tamil, but user penetration has not kept pace. Developing systems that can answer common questions about the web is a critical first step in making it even more useful. There has been very little research in this area for Hindi, and Tamil, even fewer datasets are available to the general public as a result. There is a need to develop a system that can understand and do the task of machine reading comprehension in Hindi and Tamil to expand the usage of technology to the greater number of users.

## 3.2 PROPOSED METHOD

### 3.2.1 Basic Approach:

BERT model is the basis of this experiment. It is pre-trained on large corpus of unlabeled data collected from well-known sources like Wikipedia. After the introduction of BERT in the field it revolutionizes the whole Natural language Processing field. It achieved revolutionized performance far better than previous model. After pretraining it can be fine tunned for several natural language processing tasks using the labeled data for better performance. Now BERT is not only used for English language, but the multilingual version of the BERT is also introduced to work for other languages. Fig 3.1 [1] shows the pretraining and finetuning procedures of BERT.

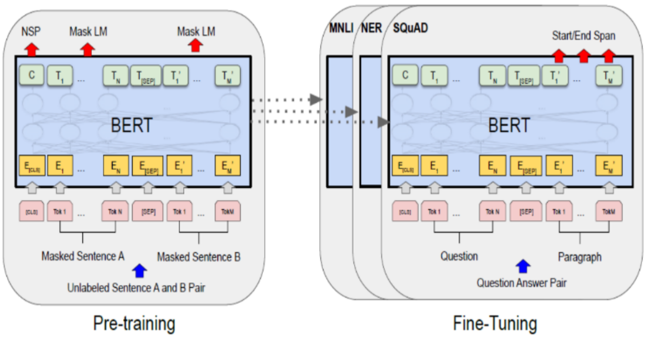
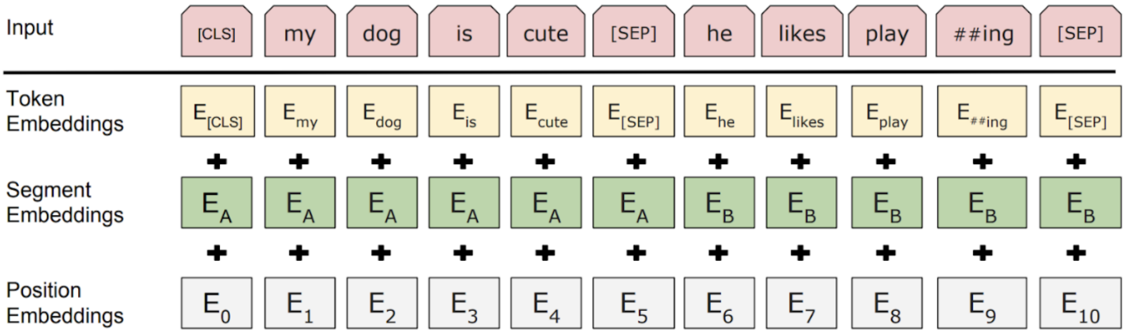


Figure 3.1. BERT Pre-training and Fine-Tuning procedures

It's that easy to understand how BERT architecture works! The encoder receives the model's input in the form of tokens, which are subsequently turned into vectors. BERT additionally needs certain information at this phase before these vectors are processed into the neural network, i.e. when you send an input phrase to the encoder, three sets of embeddings are formed.

- Token embeddings: a CLS token appears at the start of the sentence, where a SEP token (the separator token) appears at the conclusion of each sentence.
- Segment embeddings: Each word in the sentences is given a token to distinguish them as sentence A, B, or C.
- Position embeddings: assigns a unique positional token to each input token, beginning at zero, to represent the position of words in the input sequence.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | [CLS] | my | dog | is | cute | [SEP] | he | likes | play | ##ing | [SEP] |
| Token Embeddings | $E_{[CLS]}$ | $E_{my}$ | $E_{dog}$ | $E_{is}$ | $E_{cute}$ | $E_{[SEP]}$ | $E_{he}$ | $E_{likes}$ | $E_{play}$ | $E_{\#\#ing}$ | $E_{[SEP]}$ |
| | + | + | + | + | + | + | + | + | + | + | + |
| Segment Embeddings | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_B$ | $E_B$ | $E_B$ | $E_B$ | $E_B$ |
| | + | + | + | + | + | + | + | + | + | + | + |
| Position Embeddings | $E_0$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_5$ | $E_6$ | $E_7$ | $E_8$ | $E_9$ | $E_{10}$ |

Figure 3.2. BERT input representation

### 3.2.2 Our Approach:

The approach we used to train our models for this experiment is as below:

1) Take the English, Bengali and Telugu subset of TyDi dataset. Tokenize it with sequence length 256 and remove 90% of negative examples (negative sampling). Add tokenized English SQUAD and Hindi parts of MLQA and XQUAD. Shuffle all together and put aside.
2) Repeat step 1, but replace English SQUAD with subset of Natural Questions [14]. Use sequence length 384 for tokenization (progressive resizing, see below).
3) Concatenate the data collected in steps 1 and 2.
4) Train.

We used a sequential sampler and 1 epoch for training because we moved the data combination/shuffling logic into the dataset. The HuggingFace datasets library made this quite simple. We used negative sampling, usually 0.1, for TyDi, and NQ datasets because they came with longer contexts. We initially experimented with some auto-translated datasets like Tamil SQUAD but they didn't help our performance, so at the end We only used original language datasets and those translated manually (Hindi MLQA/XQUAD). We tried various languages from TyDi and found that English/Bengali/Telugu worked best. We wanted to add some more data, so we took Google Natural Questions that contained a short answer and trained a simple Roberta-base classifier to distinguish NQ data from TyDi English dataset. I selected a subset of NQ that was most similar to English TyDi. At last, we moved to a 3-fold setup for diversity and for early stopping.

### 3.2.3 Post Processing:

There was a potential to enhance predictions by addressing XLM-Roberta and Rembert tokenization difficulties, in addition to regular postprocessing (we did not edit the default routines from the official HuggingFace QA notebook). Some punctuation signs and sub-words are merged in these models, but annotators generally divide them. We started by just removing the punctuation marks. We put dots after BC/AD and parentheses if the stripped forecast is provided. We didn't conduct any postprocessing for MURIL because its tokenizer handled it effectively.

### 3.2.4 Fine Tuning:

We didn't do a lot of finetuning with different hyperparameters; we typically started with the settings specified in each backbone article and kept with them. The XLM-Roberta backbone was used for the majority of my investigations. MURIL was the most successful (We believe due to tokenization)

Here are our default hyperparameters for each backbone (batch size = 8):

- **XLM-RoBERTa:** Gradient Accumulation: 4, Learning rate: 1e-5, WD: 0.01
- **MURIL:** Gradient Accumulation: 4, Learning rate: 3e-5, WD: 0.01
- **RemBERT:** Gradient Accumulation: 16, Learning rate: 1e-5, WD: 0.01

## 3.3 MODELS USED:

- **XLM-RoBERTa:**

XLM-RoBERTa[5] is the model we utilized. XLM-RoBERTa[5] is a multilingual masked language model which is established on a transformer idea that is already pre-trained on 2.5T commonCrawl data provided in 100 languages and delivers cutting-edge cross-lingual classification and question answering performance. It provides better performance on cross-lingual task than previous multilingual models as m-BERT. In this experiment we XLM-Roberta large model which has 550M params, H = 1024, L = 24, A=16 and 250k vocab size. It applies sub-word tokenization on raw text by using sentence piece [22] method.

- **RemBERT:**

RemBERT [4] is a multilingual model which is established on the BERT system architecture that rethinks how weights are exchanged across input and output embeddings in cutting-edge pre-trained models. It shows how decoupled embeddings increase modelling flexibility by enabling for considerable improvements in parameter allocation efficiency in multilingual model input embedding. By reallocating input embedding parameters in the Transformer layers during fine-tuning, it achieves much better performance on traditional natural language comprehension tasks with the same amount of parameters. It also shows that increasing the output embedding's capacity improves the model's fine-tuning performance, despite the output embedding being removed after pre-training.

- **MURIL:**

Multilingual Representations for Indian Languages(MuRIL) [3] is an machine learning system designed preferebly for Indian languages. MuRIL now supports 17 languages (16 IN and English) [3]. It helps to build indigenous technology by providing a common foundation in vernacular languages. Its main goal was to make the internet more accessible to Indian regional languages and to improve the performance of downstream NLP activities. It also tries to address issues linked to Indian languages, such as spelling differences, transliteration, and so on. It adds translated and transliterated document pairings to monolingual text corpora, which serve as supervised cross-lingual signals in training.

# CHAPTER 4

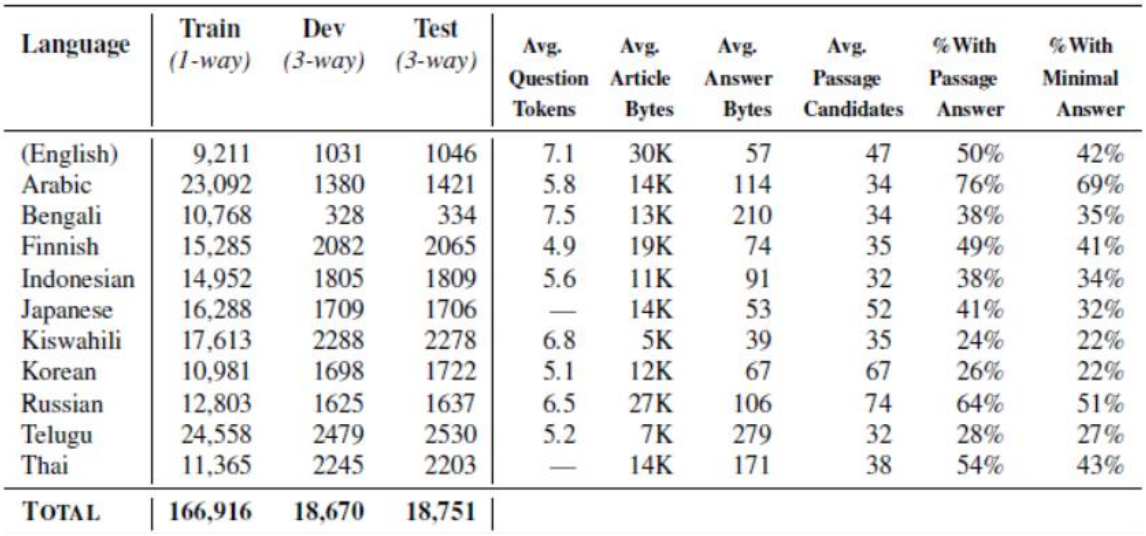
# WORKING AND ANALYSIS

## 4.1 DATASET:

As there was no single large enough dataset available for Hindi and Tamil. We collected and combined Hindi and Tamil data from different standard datasets TyDi[7], MLQA[6] and XQUAD[8]. TyDi contains 204k question-answer pairs in which there are almost 30k Telugu language pairs. MLQA dataset contains 46k question answer pairs of 7 languages in which there are almost 5k Hindi language pairs. XQUAD is a cross-lingual dataset contains 1190 question answer pairs and 240 paragraphs translated from SQUAD 1.1 in 10 languages like Hindi.

### 4.1.1 TyDi dataset:

TyDi QA is a question-answer dataset with 204K question-answer pairings that covers 11 typologically diverse languages. Because the typology of the languages in TyDi QA is so varied (the collection of linguistic traits that each language conveys), we anticipate models that perform well on this set to generalize to a large number of languages throughout the globe. It comprises linguistic phenomena that are absent from English-only corpora. To prevent priming effects and give a genuine information-seeking task, Unlike SQuAD and its progeny, here data is directly collected in each language from the people don’t know the answer of questions and want to know the same hence removing the need of language translation (unlike MLQA and XQuAD).

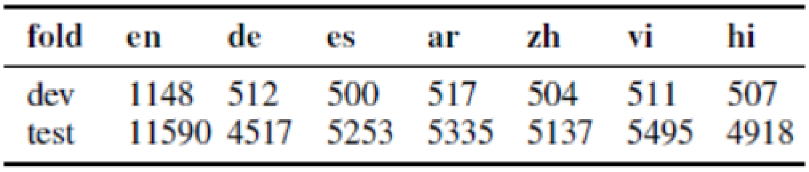


| Language | Train *(1-way)* | Dev *(3-way)* | Test *(3-way)* | Avg. Question Tokens | Avg. Article Bytes | Avg. Answer Bytes | Avg. Passage Candidates | % With Passage Answer | % With Minimal Answer |
|---|---|---|---|---|---|---|---|---|---|
| (English) | 9,211 | 1031 | 1046 | 7.1 | 30K | 57 | 47 | 50% | 42% |
| Arabic | 23,092 | 1380 | 1421 | 5.8 | 14K | 114 | 34 | 76% | 69% |
| Bengali | 10,768 | 328 | 334 | 7.5 | 13K | 210 | 34 | 38% | 35% |
| Finnish | 15,285 | 2082 | 2065 | 4.9 | 19K | 74 | 35 | 49% | 41% |
| Indonesian | 14,952 | 1805 | 1809 | 5.6 | 11K | 91 | 32 | 38% | 34% |
| Japanese | 16,288 | 1709 | 1706 | — | 14K | 53 | 52 | 41% | 32% |
| Kiswahili | 17,613 | 2288 | 2278 | 6.8 | 5K | 39 | 35 | 24% | 22% |
| Korean | 10,981 | 1698 | 1722 | 5.1 | 12K | 67 | 67 | 26% | 22% |
| Russian | 12,803 | 1625 | 1637 | 6.5 | 27K | 106 | 74 | 64% | 51% |
| Telugu | 24,558 | 2479 | 2530 | 5.2 | 7K | 279 | 32 | 28% | 27% |
| Thai | 11,365 | 2245 | 2203 | — | 14K | 171 | 38 | 54% | 43% |
| **TOTAL** | **166,916** | **18,670** | **18,751** | | | | | | |

Table 4.1. TyDi Dataset statictics

### 4.1.2 MLQA dataset:

The MultiLingual Question Answering (MLQA) dataset is a standard for assessing cross-linguistic question-answering ability. MLQA provides extractive Q-A numbered over to 5K examples (12K English samples) in SQuAD format in Arabic, Simplified Chinese, English, German, Hindi, Spanish, and Vietnamese. MLQA is very concurrent, with four Q-A instances operating in concurrent on average.

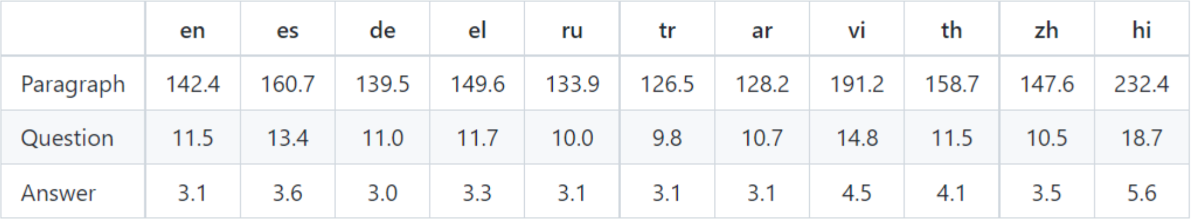


| fold | en | de | es | ar | zh | vi | hi |
|---|---|---|---|---|---|---|---|
| dev | 1148 | 512 | 500 | 517 | 504 | 511 | 507 |
| test | 11590 | 4517 | 5253 | 5335 | 5137 | 5495 | 4918 |

Table 4.2. MLQA Dataset Statistics

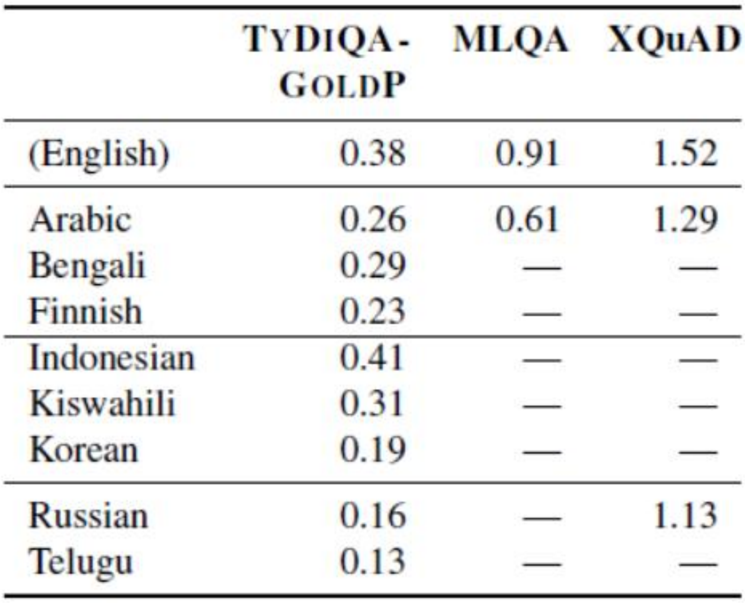
### 4.1.3 XQUAD dataset:

The Cross-lingual Question Answering Dataset (XQuAD) is a standard dataset used to evaluate cross-linguistic question-answering capacity. From the SQuAD v1.1 development set small set of 1190 question answer pairs and 240 paragraphs are included in the dataset, as well as professional translations into ten languages: Arabic, Chinese, German, Greek, Hindi, Russian, Spanish, Thai, Turkish, and Vietnamese. As a consequence, the dataset in 11 languages is totally parallel.



| | en | es | de | el | ru | tr | ar | vi | th | zh | hi |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Paragraph | 142.4 | 160.7 | 139.5 | 149.6 | 133.9 | 126.5 | 128.2 | 191.2 | 158.7 | 147.6 | 232.4 |
| Question | 11.5 | 13.4 | 11.0 | 11.7 | 10.0 | 9.8 | 10.7 | 14.8 | 11.5 | 10.5 | 18.7 |
| Answer | 3.1 | 3.6 | 3.0 | 3.3 | 3.1 | 3.1 | 3.1 | 4.5 | 4.1 | 3.5 | 5.6 |

Table 4.3. XQuAD Dataset Statistics



| | TYDIQA-GOLDP | MLQA | XQuAD |
|---|---|---|---|
| (English) | 0.38 | 0.91 | 1.52 |
| Arabic | 0.26 | 0.61 | 1.29 |
| Bengali | 0.29 | — | — |
| Finnish | 0.23 | — | — |
| Indonesian | 0.41 | — | — |
| Kiswahili | 0.31 | — | — |
| Korean | 0.19 | — | — |
| Russian | 0.16 | — | 1.13 |
| Telugu | 0.13 | — | — |

Table 4.4. For TYDIQAGOLDP, MLQA, and XQuAD, lexical overlap data reveal the average number of tokens shared between the question and a 200-character window surrounding the response span.

## 4.2 EVALUATION METHOD

### 4.2.1 F1-Score

For Machine Reading Comprehension systems evaluation harmonic mean of recall and accuracy is regular metric. It regards the system produce and the ground-truth reaction as words. Precision is often calculated by dividing the amount of correctly predicted tokens by the total amount of expected tokens. To calculate recall, divide the number of accurately predicted tokens by the set of known tokens. The following is how the F1 score is calculated:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

### 4.2.2 Accuracy:

The proportion of questions that an Machine Reading Comprehension system correctly answers is known as accuracy. Consider the following scenario: an Machine Reading Comprehension task has N number of questions, each question has one valid answer, the responses may be a word, a paragraph, or a sentence, suppose the system properly answers M questions. The following is the accuracy equation:

$$\text{Accuracy} = \frac{M}{N}$$

### 4.2.3 Exact Match (EM)

Some of the terms in the system-generated solution may be genuine answers, while others may not. Exact Match in this example refers to the proportion of questions where the responses generated by system response matches the right answer perfectly, meaning every word is identical. The acro0nym EM stands for Exact Match. For example, if a Machine Reading Comprehension task has total N number of questions, each question has one valid answer, the responses might be a word, a phrase, or a sentence, suppose the system correctly answers M questions, the total number of questions is M. Some of the remaining N – M responses may include some genuine response phrases, but they may not match some genuine response answer entirely. Following that, the EM may be determined as below:

$$\text{Exact Match} = \frac{M}{N}$$

As a result, EM and Accuracy are the same for the prediction job of span. EM is seldom utilized in a multi-choice job since there is never a case when the response contains a piece of the right answer. It is also typical to gather numerous right responses for each question to make the assessment more credible. As a result, the precise match score is only necessary if any of the correct responses match.

### 4.2.4 Mean Average Precision (MAP)

When the system produces several replies with ratings, this metric is employed. The mean of each pair's Average Precision scores (AveP) is the Mean Average Precision for a question-answer pairings collection. Q is the total number of inquiries. AveP is a metric for evaluating information retrieval systems. In answer to a query, it analyses a ranked list of documents. The ranked list of replies for a specific question is assessed in MRC literature. The precision’s average during the span in precision-recall curve ranging from recall=0 to recall=1 is determined as AveP.

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q},$$

### 4.2.5 Mean Reciprocal Rank:

TREC QA track 1999 presented this as a standard assessment criterion for factual QA systems. MRR assesses a ranked list of responses depend on the right answer’s inverse of the rank score, according to the definition described in the "Evaluation of Factoid Answers" Section of the "Speech and Language Processing" book. For instance, if the right answer in a system's output list has a rank of 4, the question’s reciprocal rank score is 1/4. Then value is averaged throughout the whole test set of questions.

# CHAPTER 5

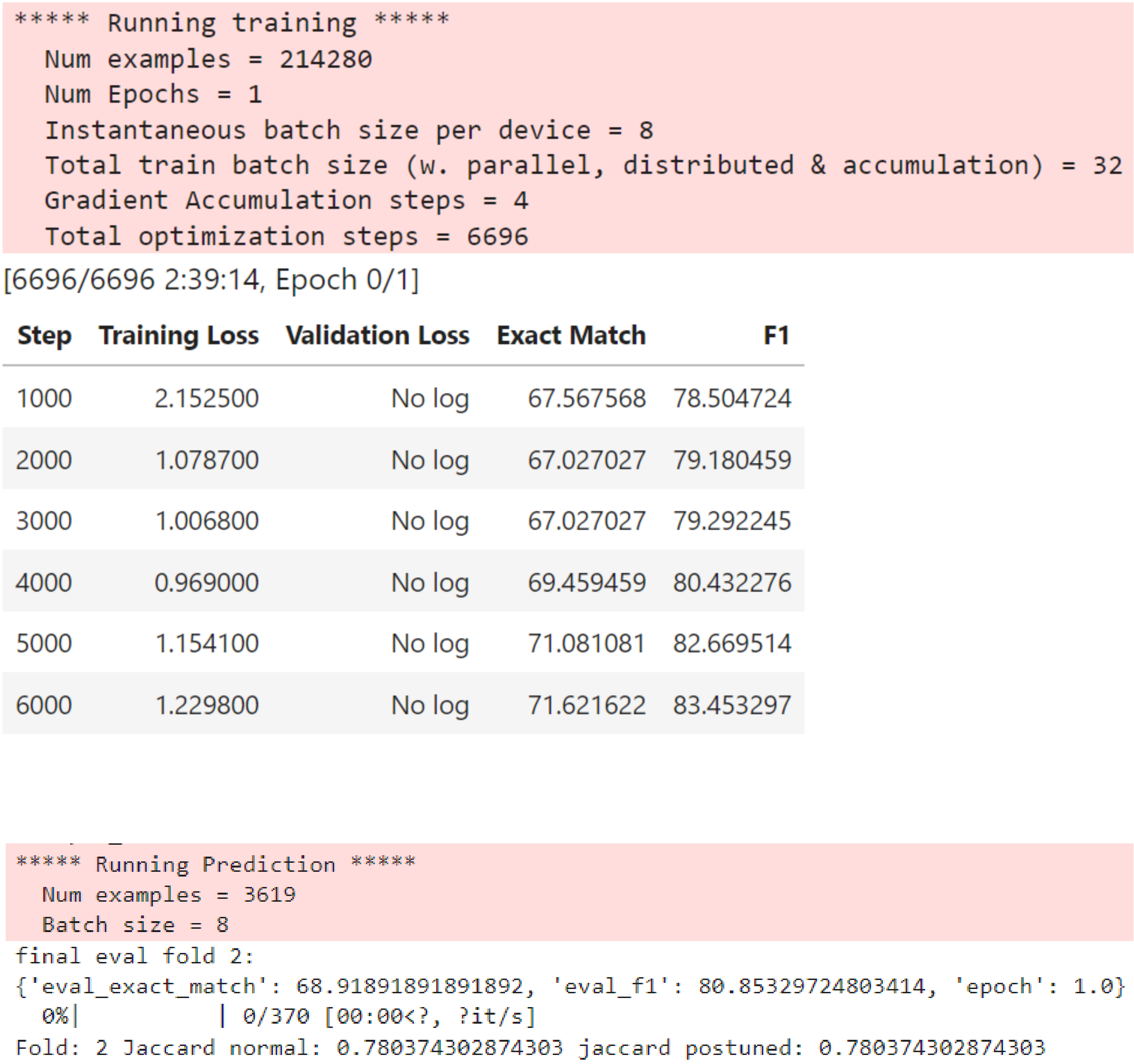
# EXPERIMENTS AND RESULTS

## 5.1 Training results and comparison

In our work we worked upon some of the major multilingual pre-trained models based on transformer. We used different standard datasets like XSQUAD, MLQA, TyDi to extract data to train and analyze our work. Collectively they contain around 250k question answer pairs from which used data of some specific languages like Hindi, Tamil, Bengali, and English.

The F1 score, Exact match, and Jaccard Index were used to evaluate the performance. The model's performance for the actual positive class will be shown via precession. The recall will tell us how well our model performed for positive cases in general. Precession and recall will be balanced using the F1-score. The accuracy of our model will tell us how exact it is. Jaccard index is also referred as similarity index which is used to find out the similarity between two sets what are shared between them and what are different. It measures the similarity between two sets in the range of 0-100%.

The performance results of the models we have trained are shown in Table 5.1, MURIL-Large, RemBERT-Large, and XLM-RoBERTa-Large. From the results we can we that the performance of all the models were promising, RemBERT achieved the best F1 score, Exact Match and Jaccard Index of 84.58%, 74.05%, and 0.81 respectively for the used dataset. The performance comparison of all the models we have trained in this experiment is shown in Fig 5.4.



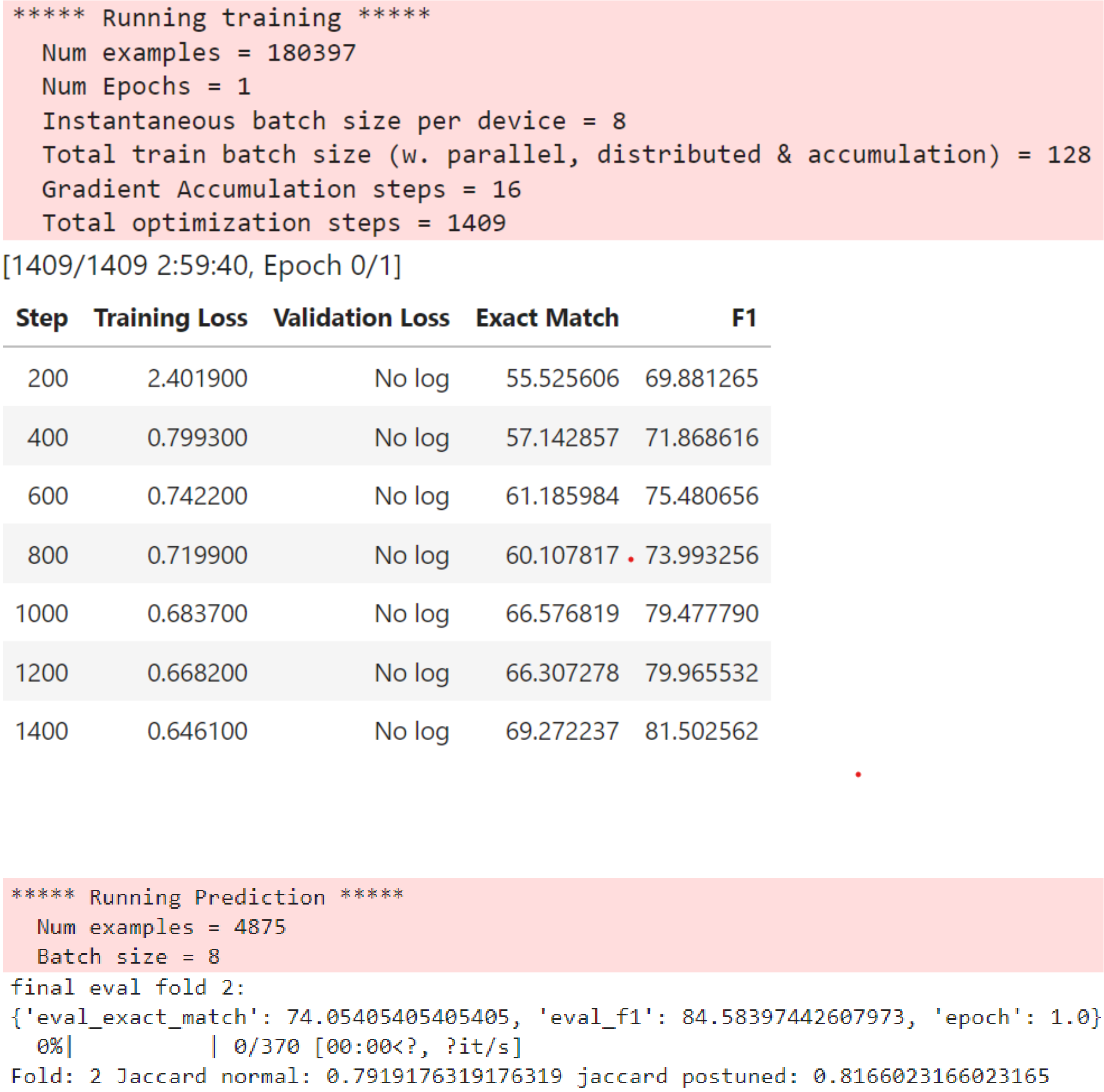
```
***** Running training *****
  Num examples = 214280
  Num Epochs = 1
  Instantaneous batch size per device = 8
  Total train batch size (w. parallel, distributed & accumulation) = 32
  Gradient Accumulation steps = 4
  Total optimization steps = 6696
```

[6696/6696 2:39:14, Epoch 0/1]

| Step | Training Loss | Validation Loss | Exact Match | F1 |
|---|---|---|---|---|
| 1000 | 2.152500 | No log | 67.567568 | 78.504724 |
| 2000 | 1.078700 | No log | 67.027027 | 79.180459 |
| 3000 | 1.006800 | No log | 67.027027 | 79.292245 |
| 4000 | 0.969000 | No log | 69.459459 | 80.432276 |
| 5000 | 1.154100 | No log | 71.081081 | 82.669514 |
| 6000 | 1.229800 | No log | 71.621622 | 83.453297 |

```
***** Running Prediction *****
  Num examples = 3619
  Batch size = 8
final eval fold 2:
{'eval_exact_match': 68.91891891891892, 'eval_f1': 80.85329724803414, 'epoch': 1.0}
  0%|          | 0/370 [00:00<?, ?it/s]
Fold: 2 Jaccard normal: 0.780374302874303 jaccard postuned: 0.780374302874303
```

Fig 5.1. MuRIL model training results



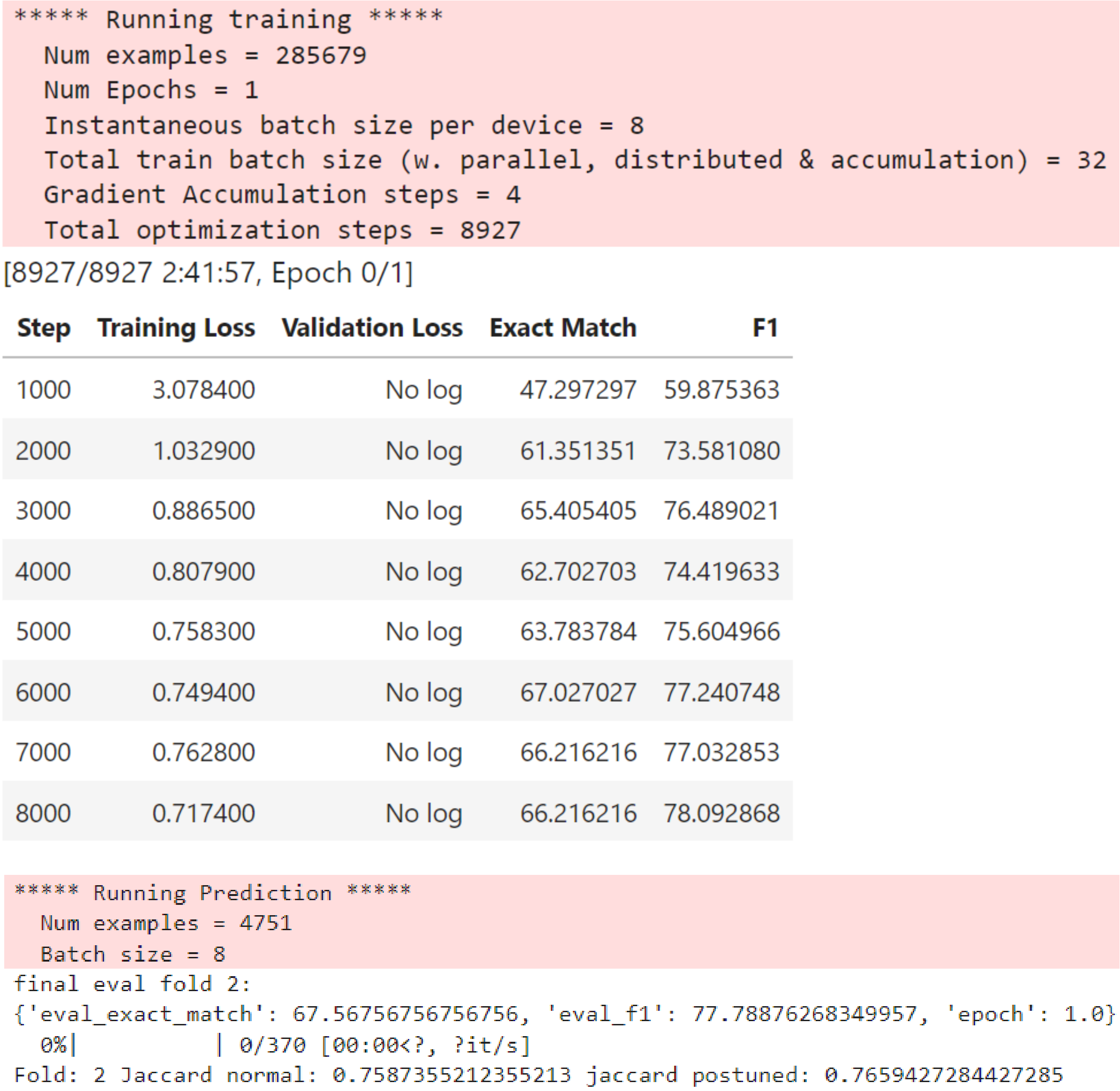
```
***** Running training *****
  Num examples = 180397
  Num Epochs = 1
  Instantaneous batch size per device = 8
  Total train batch size (w. parallel, distributed & accumulation) = 128
  Gradient Accumulation steps = 16
  Total optimization steps = 1409
```

[1409/1409 2:59:40, Epoch 0/1]

| Step | Training Loss | Validation Loss | Exact Match | F1 |
|---|---|---|---|---|
| 200 | 2.401900 | No log | 55.525606 | 69.881265 |
| 400 | 0.799300 | No log | 57.142857 | 71.868616 |
| 600 | 0.742200 | No log | 61.185984 | 75.480656 |
| 800 | 0.719900 | No log | 60.107817 | 73.993256 |
| 1000 | 0.683700 | No log | 66.576819 | 79.477790 |
| 1200 | 0.668200 | No log | 66.307278 | 79.965532 |
| 1400 | 0.646100 | No log | 69.272237 | 81.502562 |

```
***** Running Prediction *****
  Num examples = 4875
  Batch size = 8
final eval fold 2:
{'eval_exact_match': 74.05405405405405, 'eval_f1': 84.58397442607973, 'epoch': 1.0}
  0%|          | 0/370 [00:00<?, ?it/s]
Fold: 2 Jaccard normal: 0.7919176319176319 jaccard postuned: 0.8166023166023165
```

Fig 5.2. RemBERT model training results



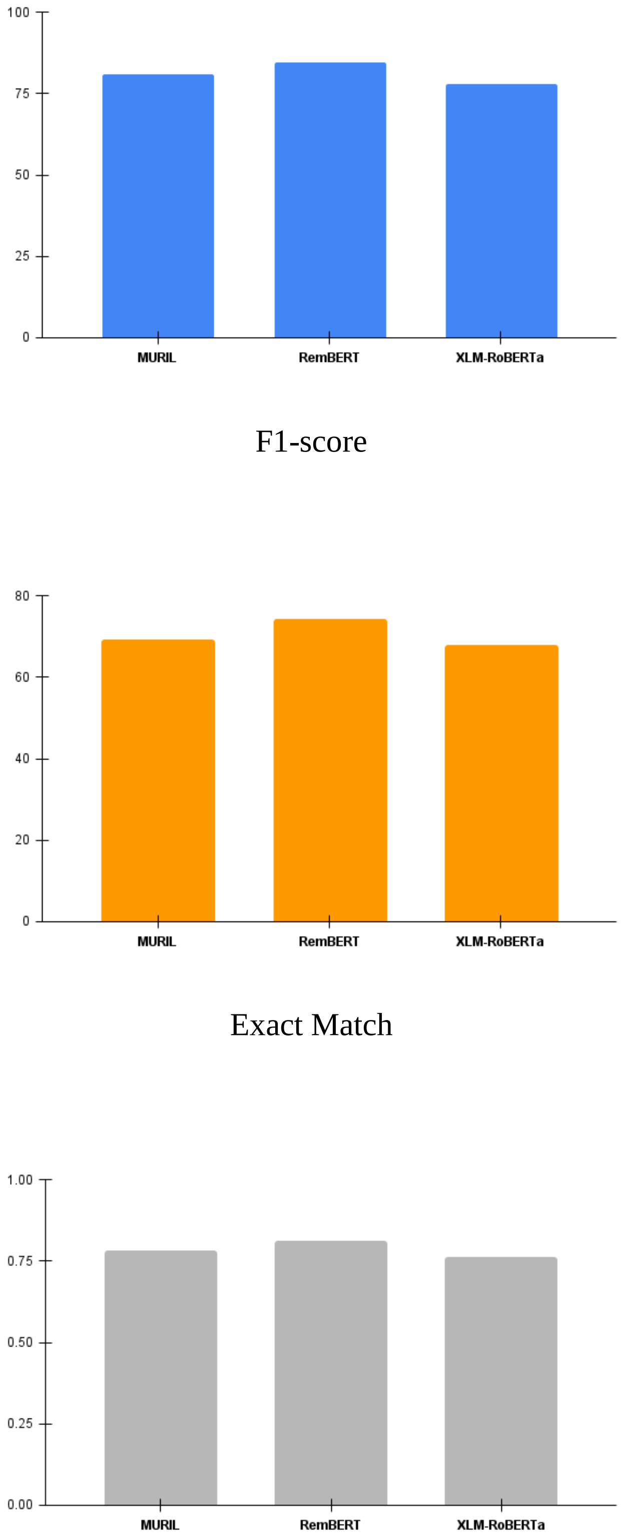
```
***** Running training *****
  Num examples = 285679
  Num Epochs = 1
  Instantaneous batch size per device = 8
  Total train batch size (w. parallel, distributed & accumulation) = 32
  Gradient Accumulation steps = 4
  Total optimization steps = 8927
```

[8927/8927 2:41:57, Epoch 0/1]

| Step | Training Loss | Validation Loss | Exact Match | F1 |
|---|---|---|---|---|
| 1000 | 3.078400 | No log | 47.297297 | 59.875363 |
| 2000 | 1.032900 | No log | 61.351351 | 73.581080 |
| 3000 | 0.886500 | No log | 65.405405 | 76.489021 |
| 4000 | 0.807900 | No log | 62.702703 | 74.419633 |
| 5000 | 0.758300 | No log | 63.783784 | 75.604966 |
| 6000 | 0.749400 | No log | 67.027027 | 77.240748 |
| 7000 | 0.762800 | No log | 66.216216 | 77.032853 |
| 8000 | 0.717400 | No log | 66.216216 | 78.092868 |

```
***** Running Prediction *****
  Num examples = 4751
  Batch size = 8
final eval fold 2:
{'eval_exact_match': 67.56756756756756, 'eval_f1': 77.78876268349957, 'epoch': 1.0}
  0%|          | 0/370 [00:00<?, ?it/s]
Fold: 2 Jaccard normal: 0.7587355212355213 jaccard postuned: 0.7659427284427285
```

Fig 5.3. XLM-RoBERTa model training results



F1-score

Exact Match

Jaccard Index

Fig 5.4. Performance comparison of models

| Models | F1-Score | Exact Match | Jaccard Index |
|---|---|---|---|
| MURIL | 80.85 | 68.91 | 0.78 |
| RemBERT | 84.58 | 74.05 | 0.81 |
| XLM-RoBERTa | 77.78 | 67.56 | 0.76 |

Table 5.1. Performance evaluation of models

# CHAPTER 6
# Future Opportunities

In recent times, the Machine Reading Comprehension mission has achieved significant progress. In the MRC problem, BERT language models are being fine-tuned[1] and XLNet on the target job has shown amazing results, to the point that many cutting-edge technologies rely on these language models. They do, however, have several flaws that prevent them from achieving true reading comprehension. Some of these issues and emerging developments in the MRC sector are listed below:

Distributions outside of the domain: MRC models are too brittle for out-of-domain dispersed data, despite their great accuracy on test samples from their training distribution. Some recent publications have focused on increasing the generalization capabilities of MRC based models to solve this kind of problem.

Multi-document MRC: multi-hop reasoning is Among the most difficult aspects of the MRC problem is, which requires inferring an answer from many documents. These texts may be lot of paragraphs from a single document or heterogeneous paragraphs from various papers. One of the newest ideas is to employ graph structures for multi-hop reasoning, such as neural networks based on graph.

Mathematical reasoning: Many issues in real-world applications, such as addition, subtraction, and comparison, need numerical inference. Consider the query from the DROP dataset that requires a subtraction: "How much more did the Untitled (1981) artwork sell for than the 12 million dollar estimate?" In recent years, The development of MRC models capable of numerical reasoning is becoming more prevalent, particularly with the emergence of numerical datasets like DROP.

No-Answer Questions: The capacity for models to detect questions that can't be answered in the present environment is one of the important advancements that makes MRC systems more accessible in real-world applications. More attention has been dedicated to this problem with the introduction of datasets including these types of questions pairs, such as Natural Questions [14] and SQuAD 2.0.

Non-factual inquiries: Non-factual inquiries, many inquiries, such as why and opinion, need the development of responses rather than the selection of a context span. The present models' accuracy in addressing These concerns are still unsatisfactory. In recent times, a number of datasets including non-factual questions have received increased interest in this kind of inquiry.

Low-resource languages datasets and models: It's worth mentioning that the English and Chinese are two languages with a lot of data. A recent trend in this discipline is the creation of new datasets and models for languages with limited resources and their development in a multilingual or multi-task scenario.

# CHAPTER 7

# CONCLUSION

We explored at several deep learning models for Multilingual Machine Reading Comprehension based on transformer in this study. After testing with MURIL-Large, RemBERT-Large, and XLM-RoBERTa-Large DNN models, With an F1 score of 84.58, an Exact Match of 74.05, and a Jaccard Index of 0.81, RemBERT-Large outperformed the other models for the given datasets. If we look at the findings closely, we can observe that none of the models show a significant difference between Exact Match and F1-Score. I didn't alter with the HuggingFace Library's default tokenizers, but I did try splitting certain punctuation marks before running the tokenizer, which didn't boost my score either. Our model's drawback is that we only trained it for two of the Indian languages (Hindi, and Tamil), and the data supplied for this experiment was insufficient. Our technique is further limited by the fact that the model only works for English and two Indian languages, Hindi and Tamil.

We may use these models to train for more Indian languages in the future, as well as train them for other elements of machine reading comprehension.

# REFERENCES

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] Gupta, S., & Khade, N. (2020). Bert based multilingual machine comprehension in english and hindi. arXiv preprint arXiv:2006.01432.

[3] Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., ... & Talukdar, P. (2021). Muril: Multilingual representations for indian languages. arXiv preprint arXiv:2103.10730.

[4] Chung, H. W., Fevry, T., Tsai, H., Johnson, M., & Ruder, S. (2020). Rethinking embedding coupling in pre-trained language models. arXiv preprint arXiv:2010.12821.

[5] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

[6] Lewis, P., Oğuz, B., Rinott, R., Riedel, S., & Schwenk, H. (2019). MLQA: Evaluating cross-lingual extractive question answering. arXiv preprint arXiv:1910.07475.

[7] Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., & Palomaki, J. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics, 8, 454-470.

[8] Artetxe, M., Ruder, S., & Yogatama, D. (2019). On the cross-lingual transferability of monolingual representations. arXiv preprint arXiv:1910.11856.

[9] Soares, M. A. C., & Parreiras, F. S. (2020). A literature review on question answering techniques, paradigms and systems. Journal of King Saud University-Computer and Information Sciences, 32(6), 635-646.

[10] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[11] Kexin, X. (2021). Literature Review on Neural Machine Comprehension for Question Answering.

[12] Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291

[13] Pathak, A. (2021, January). Comparative Analysis of Transformer based Language Models. In CS & IT Conference Proceedings (Vol. 11, No. 1). CS & IT Conference Proceedings..

[14] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., ... & Petrov, S. (2019). Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7, 453-466.

[15] Liu, S., Zhang, X., Zhang, S., Wang, H., & Zhang, W. (2019). Neural machine reading comprehension: Methods and trends. Applied Sciences, 9(18), 3698.

[16] Zhang, Z., Yang, J., & Zhao, H. (2020). Retrospective reader for machine reading comprehension. arXiv preprint arXiv:2001.09694, 1, 1-9.

[17] Aniol, A., Pietron, M., & Duda, J. (2019, November). Ensemble approach for natural language question answering problem. In 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW) (pp. 180-183). IEEE.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[19] Singh, J., McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2019). Xlda: Cross-lingual data augmentation for natural language inference and question answering. arXiv preprint arXiv:1905.11471.

[20] Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. arXiv preprint arXiv:1806.03822.

[21] Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.

[22] Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

[23] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[24] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410.

# LIST OF PUBLICATIONS

**[1]** Shailender Kumar, Deepak, “BERT-based models' impact on machine reading comprehension in Hindi and Tamil”. Accepted **at the 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)**

Indexed by Scopus, EBSCO, ProQuest, ULRICH, J-Gate, OAJI**.**
Paper Id: IJAST_06_2020_507

**Abstract-** “Multilingual Machine Comprehension" is a QA sub-task that comprises citing an answer to a question from a context, even if that answer written in a separate language from the excerpt itself. A lot of models have been trained to answer the question from a given short context which is a limitation of MRC, few models are considering this problem and adapting to handle the large input context to make the MRC more accessible and applicable to open domain scenarios. In this study, we examine Multilingual Representations for Indian Languages (MuRIL), rebalanced multilingual BERT (RemBERT), and XLM-RoBERTa, which are all BERT-based deep learning models. We trained these models to work on multilingual MRC particularly for two of the most used Indian languages Hindi and Tamil The datasets utilized in this study are freely available. The results of our research reveal that RemBERT outperformed other BERT-based deep learning models. For the dataset employed, the model received an F1 score of 84.58, an Exact Match of 74.05, and a Jaccard Index of 0.81.