Question Answering in Biomedical Domain

By

Malita Dodti

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2022

Question Answering in Biomedical Domain

By

Malita Dodti

APPROVED BY:

____

H. Zhang

Department of Biomedical Sciences

____

L. Rueda

School of Computer Science

____

J. Chen, Advisor

School of Computer Science

December 8, 2022

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Question Answering (QA) is a complex Natural Language Processing (NLP) task. It involves understanding a question, retrieving relevant materials, and generating a suitable answer. Its major challenge is to create proper representations of the language and to produce a suitable answer to a given question. Pretraining neural language models has significantly improved many natural language processing tasks. In particular, BERT is a deeply bidirectional, pre-trained language representation that has performed well in NLP tasks including question answering. In this thesis work, we study the application of the BERT technique to automated response generation for biomedical text mining. This application comes from the consideration that, due to the growth of the volume of biomedical papers, biomedical text mining is demanding better techniques to automate the extraction and the summarization of the biomedical information and to automate the responses to the queries. To be successful in answering biomedical questions, the lack of the availability of large expert-annotated biomedical datasets must be addressed. In the present thesis work, we consider augmenting the data samples from existing ones by varying context lengths. We have studied how dynamic changes in the passage length affect the performance of the models. This provides us with a better understanding of the optimal context lengths. To learn about the behaviour of the models when unanswerable questions are present, datasets with various ratios of answerable and unanswerable questions are used and the experiments show a significant range of the behaviour of the prediction models on different training and testing sets. During the experiments, a new span selection technique is implemented for predicting the answers. According to the experiments, it offers satisfactory improvement to the effectiveness of the state-of-the-art techniques for performing question-answering tasks in the context of biomedical text mining.

DEDICATION

TO MY PARENTS

Mr. Michael Dodti and Mrs. Eliza Dodti

With love and eternal appreciation.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my thesis advisor, Dr. Jessica Chen for the continuous support of my Master study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

I would also like to thank Dr. Luis Rueda and Dr. Huiming Zhang for taking time to review my thesis and provide valuable feedback on my thesis. I would like to express my gratitude towards Mrs. Christine Weisener for providing me with support and assistance in academic matters.

I would like to thank my parents for their continuous motivation and support throughout the journey towards a master's degree. Your prayers for me were what sustained me this far. My family and friends for all their blessings.

Special thanks to my husband Mr. Denzing Pen for being understanding and encouraging when undertaking my research and writing my thesis.

Finally, I would like to thank God, for letting me through all the difficulties. I have experienced your guidance day by day. I will keep on trusting you for my future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Question Answering

The introduction of Question Answering Systems (QAS) is a viable option and an effective strategy for finding important information online. QA systems provide an answer to every question by finding brief text passages or phrases from numerous documents, including the actual answer. QA systems have been developed for various domains, databases, question types, and answer formats [7]. Academic research has achieved essential advancements that have allowed QA to cope with various question types, such as fact, definition, list, hypothetical, how, why, cross-lingual, and semantically limited questions [42]. The domain coverage of QAS is divided into two categories: closed-domain, which explores domain-specific knowledge typically structured into ontologies, and open-domain, which relies on general ontologies and global expertise to address practically any issue. The critical difficulties in NLP for QA include speech recognition, sentiment analysis, information extraction, text summarization, and natural language generation. QAS with the help of primary sources in addition to knowledge bases and diverse corpora, will enable the user to receive a clear and accurate response to their natural language query.

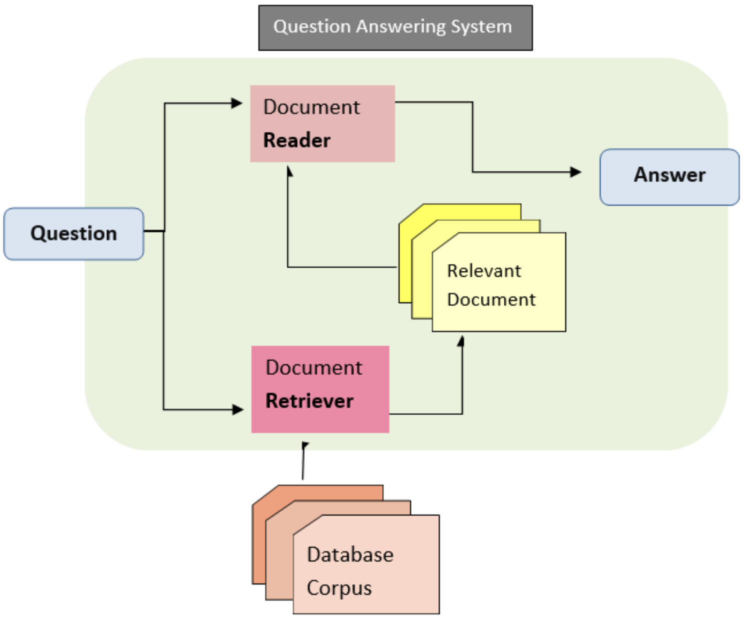


Figure 1.1: Architecture of Question Answering System

Question Answering (QA) in information retrieval (IR) and natural language processing (NLP) is the process of automatically responding to a question posed in natural language by a human. Question analysis, document retrieval, and answer extraction are the three main subtasks that make up the task of Question Answering (QA) *Figure 1.2*. Before responding to questions submitted to QA systems, they must be examined and understood. Focusing on the semantic information, constraints, and essential keywords for the QAS, the query and output representation significantly contributes to the QAS [43]. The question analysis stage is processed in various ways. Tokenization, ambiguity resolution, logical forms, semantic role labels, question restructuring, co-reference solution, relation extraction, and named entity recognition are a few of these [7]. Many QA systems have merged and employed various techniques to fit better the type of data source they are working with or to address a specific research issue.

The queries created in the question analysis module are used in the document retrieval stage to search across information sources for appropriate responses to the questions posed. The structure of the retrieved documents is based on three steps: retrieval, processing, and ranking [44]. The process of retrieving is then completed by matching documents to query patterns. The resulting texts are then ranked using ranking algorithms like tf-idf (term frequency-inverse document frequency). The next step of a Question Answering system is answer extraction. It uses the passages generated during the document extraction to produce the precise answer. It first generates a set of candidate answers from the retrieved passages and then ranks the responses using a few scoring methods. Various answer extraction approaches, such as n-grams, patterns, named entities, and syntactic structures, have been studied in earlier works [7].

Deep learning approaches have proved to be promising in natural Language processing. Models like Recurrent Neural Networks (RNN) are Feed Forward neural networks. RNN deals with sequence data where input is in defined order. RNN has several architectures (i)Vector-Sequence Models- (Example: Image captioning) the input is the vector representation of an image, and the output is the sequence sentence describing the image(ii) Sequence-Vector Models- (Example Sentiment Analysis). The movie review is an input, and a fixed size vector is the output indicating how good or bad this movie was. (iii) Sequence-Sequence Models- (Example Language Translation) Input could be a sentence in French and output is the translation in English language. However, RNNs are slow to train, and truncated version of back propagation is used to train. Also, RNNs cannot deal with long sequences very well. The gradients vanish and explode if the network is too long. LSTM network in 1991 introduced a long, short term memory cell that replaces dull neuron. This cell has a branch that allows passed information to skip a lot of the processing of the current cell and move on to the next.

Thus, allowing memory to be retained for longer sequences. Even though the LSTM overcome the longer sequence challenge to a certain level they are slower to train than RNNs because they are complex in structure. The sequential flow of information in LSTM is not well equipped with parallel computing.

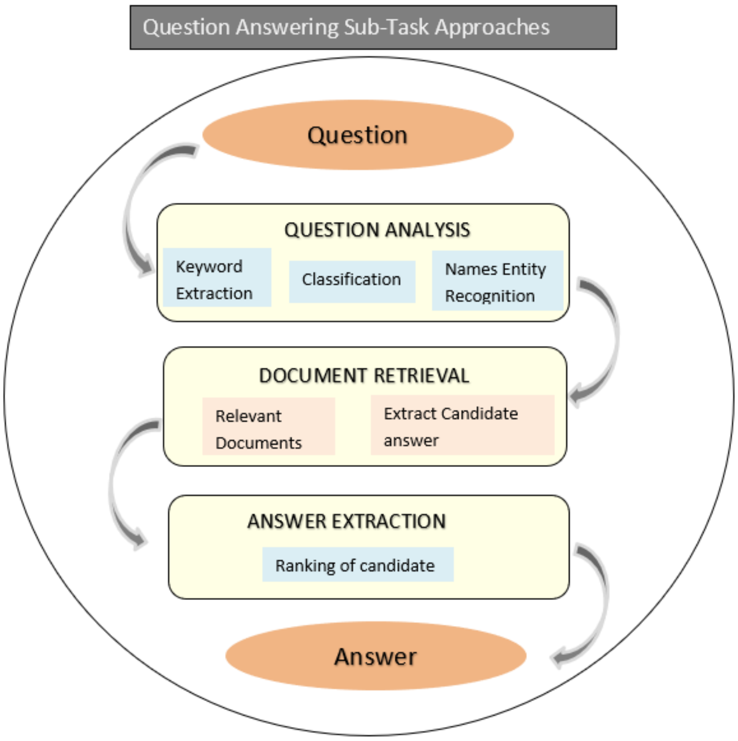


Figure 1.2: Question Answering Sub-Task Approaches

In 2017 the transformer neural network architecture was introduced. The network employs an encoder-decoder architecture like RNN, the difference is that there is no concept of the timestamp as we pass all the input simultaneously and determine the word embedding at the same time. For a better understanding of how transformers work, consider an English-to-French translation. The English sentence passes through the input embedding and the positional encoding, and we get word vectors that have positional information which is the word context. Next, the input goes through an encoder block where it passes a multi-headed attention layer and a feed-forward layer. Attention is used to determine what part of the input should be focused I.e., it determines the relevance of each word with itself and other words in the sentence. Thus, the attention vector is computed for each word to capture the contextual relationships among the words in a sentence. The simple feed-forward network transforms the attention vector into digestible input for the following block. While decoding the word vector and positional vector are given as input to the decoder block to get a notion of the context of the word in the sentence. The decoder block consists of (i) a self-attention layer which is similar to the one in the encoder block, (ii) an encoder-decoder attention block which has one vector from every word in English and French sentences capturing the main English to French word mapping and (iii) Feed-forward network. The decoder block is executed multiple times until the end of the sentence is reached.

In transformers we could separate the encoder and decoder block based on their functionality. While language translation encoder learns the grammar and context of English and decoder captures the relation of English words with French words. Both blocks understand language independently and could be used separately. If we stack encoders, we get BERT (Bidirectional Encoder Representation from Transformers). BERT can be used on down streaming task like Sentiment Analysis, Text Summarization, Question answering and many more. All these tasks require language understanding; thus, we can use pretrain BERT to understand language and finetune BERT to learn specific task depending on the problem we want to solve. BERT learns language by training on two unsupervised tasks simultaneously. They are (i) Masked language modelling and (ii) Next sentence prediction where BERT learns about bi-directional context within a sentence and sentence dependency on one another. Finetuning BERT on Question Answering is done by performing supervised training using a question answering dataset. The output parameters are learned from scratch and the rest of the model parameter are slightly finetuned. As a result, the training time is shorter. BERT finetuned on SQuAD (Stanford Question & Answer Dataset) has achieved great success in Question Answering task. BERT achieves 93.2% F1 score (a measure of accuracy), surpassing the previous state-of-the-art score of 91.6% and human-level score of 91.2%.

Most of the Question Answering approaches have been focused on the Open domain, and the challenges harbouring domain specific data particularly the biomedical field have not been well addressed. BERT success on SQuAD for Question answering paved way for researcher to use BERT model on biomedical data. A model like BioBERT pretrained on PubMed Abstract from scratch was the first step in the direction. It increased models understanding towards biomedical terms and language used. Additionally, model like Clinical Bert trained on clinical notes of patients was beneficial in addressing clinical question. UmlsBERT which incorporated domain knowledge base to better accommodate the biomedical synonyms is proved quite a significant advancement. PubMed Bert, the first model to be trained on PubMed abstracts and articles using biomedical vocabulary showed quite an improvement in the previous results. Although there are improvements in Biomedical question answering results, there are yet few challenges to overcome.

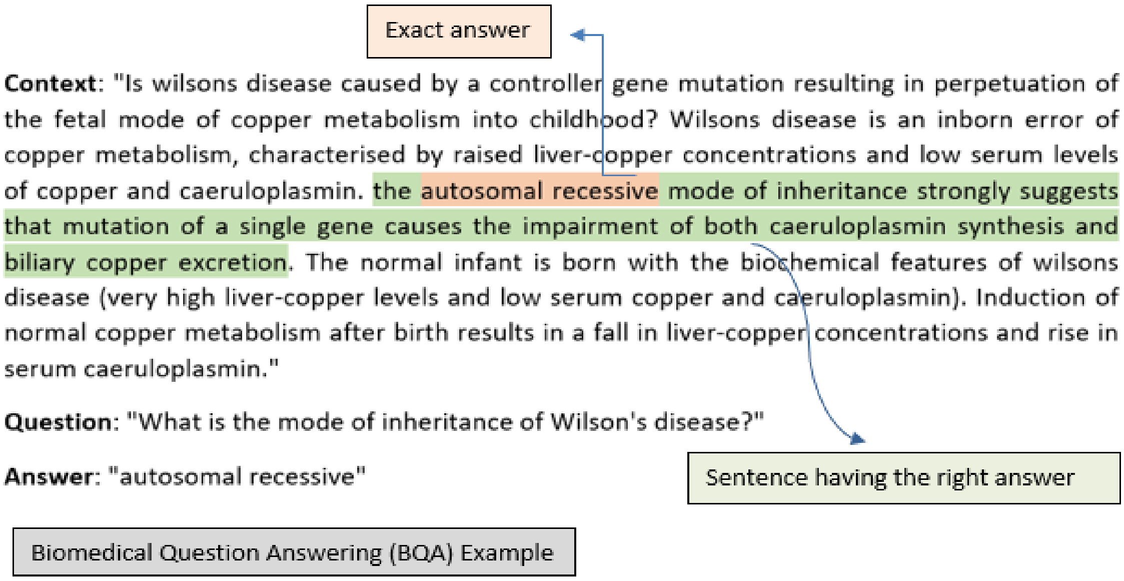
The BERT model performs better on large datasets like SQuAD, but the biomedical domain lacks large-scale expert annotated QA datasets suitable for BERT finetuning. Thus, the first step in resolving the problem is to transfer learning from SQuAD to the Biomedical dataset. To better leverage our available supervised data, we explore the guidelines in data augmentation and data sampling strategies. To augment the existing biomedical dataset, we focused on creating multiple examples of varying context length (i.e., average number of words present in the context). This increased the sample size for training the model and assisted us in identifying model performance on a variety of examples with varying context lengths. While creating examples with varying context lengths, we came across negative samples, which did not contain the answer to the question. To address this, we added negative segments to the dataset, causing it to grow larger. In the SQuAD dataset, a similar attempt was made to include a No Answer option in extractive question answering. By incorporating negative samples, the model has the option to abstain from selecting a span in a given context.

The goal was to expand the existing dataset without interfering with the model's operation, thereby improving prediction accuracy. To assess model performance, we fine-tuned the model on two dataset types, first with datasets of varying context length and then with datasets of positive and negative samples. Further investigation revealed that the span selection technique required more time to predict the start and end tokens of the answer prediction. We proposed a new span selection technique that, when compared to the previous technique, executes faster and improves the start and end token prediction of an answer.

1.2 Biomedical Domain

To help healthcare professionals find answers to medical questions, advancements in question answering (QA) is bringing the globe to new technological heights, particularly in the medical sector. The rapidly expanding biomedical literature available to medical professionals online has facilitated recent developments in biomedicine, electronic publication, and computing technology. The field of biomedicine is comprehensive and includes several biological and medical sciences. We define several biomedical subfields, including scientific, clinical, consumer health, and examination.

Scientific QA focuses on pressing issues whose solutions must be deduced or retrieved from scientific literature, for example, the Name synonym of Acrokeratosis paraneoplastica. Most recent discoveries in the biomedical sector are published in scientific literature, whose volume is expanding at a record-breaking rate. It is difficult to manually read all pertinent research and provide thorough responses to scientific queries, making robotic replying to scientific questions essential. A notable demonstration of scientific QA is the BQA community's battle against COVID-19. Large-scale corpora like PubMed and PubMed Central, each including 4.5 billion and 13.5 billion tokens respectively, are openly accessible and stand out as the most distinguishing aspect of scientific BQA. The semi-structured documents in PubMed and PubMed Central, including sections for background, introduction, methodology, and conclusion, make them potentially useful for creating domain-specific datasets.



Context: "Is wilsons disease caused by a controller gene mutation resulting in perpetuation of the fetal mode of copper metabolism into childhood? Wilsons disease is an inborn error of copper metabolism, characterised by raised liver-copper concentrations and low serum levels of copper and caeruloplasmin. the autosomal recessive mode of inheritance strongly suggests that mutation of a single gene causes the impairment of both caeruloplasmin synthesis and biliary copper excretion. The normal infant is born with the biochemical features of wilsons disease (very high liver-copper levels and low serum copper and caeruloplasmin). Induction of normal copper metabolism after birth results in a fall in liver-copper concentrations and rise in serum caeruloplasmin."

Question: "What is the mode of inheritance of Wilson's disease?"

Answer: "autosomal recessive"

Exact answer
Sentence having the right answer
Biomedical Question Answering (BQA) Example

Figure 1.3: Biomedical Question Answering Example

Clinical Question Answering focuses on addressing concerns raised by healthcare professionals regarding patients' access to medical care. It deals with specialized language disparities between clinical narratives (such as doctor notes) and non-clinical biomedical texts. All patient health information is stored in Electronic Medical Records in both organized (tables) and unstructured (medical notes) formats. Electronic Medical Records (EMRs) for the patients whose questions are explicitly addressed should be made available. For the doctors to manually check the EMRs for clinical questions regarding the patient because of the complexity and size of the EMR data would take too much time and be inefficient. Clinical QA systems can swiftly and reliably answer these inquiries to address such information needs. The biggest obstacle facing the development of clinical BQA systems is the absence of extensive expert-annotated datasets that accurately reflect the clinic's needs. Additionally, there are ethical and privacy concerns with disclosing clinical notes, mainly when the databases are based on EMRs.

The general public frequently asks about consumer health issues on search engines, where online medical services provide customers significant convenience because they are not constrained by time and geography. Before seeing a doctor or deciding whether to see a doctor, many people try to discover solutions to their medical issues. Their information demands range from self-diagnosis to locating drugs. Because customers cannot assess the calibre of medical contents, it is imperative to provide truthful responses to such inquiries. Given the conflict between the high demand from customers and the scarcity of medical professionals, an automated responding system helps distribute helpful resources to offer online medical services. Finding the appropriate or accepted responses are required for these consumer BQA datasets. Since the replies are submitted by members of online communities and the forum data contains inherent uncertainty, the quality of such datasets is in doubt. Consumer health QA faces additional obstacles because most consumers lack expertise in the biomedical field. On the other hand, the returned responses must be both correct and understandable.

1.3 Problem Definition

Biomedical Question Answering employs span selection technique to predict the answer's start and end token positions, for extractive answer prediction. The existing span selection technique consumed a lot of time because it selects the top 20 most likely start and end positions, creates 20*20 start-end position pairs, ignores invalid pairs, and selects the top pair with the highest average value. The problem considered in this thesis work is to look at suitable ways to optimise the span selection technique which computes faster and improves the answer prediction.

The BERT model performs better on large datasets. BioASQ, which is the largest available expert-annotated biomedical dataset, is twenty times smaller than SQuAD, and creating a dataset like this is expensive. Hence, data augmentation and data sampling techniques are employed to increase the size of available expert annotated dataset. These can be achieved by dividing the existing context into multiple segments in order to introduce multiple examples to the dataset. We are interested in knowing how to make the division of the context, particularly in terms of the length of the context, in order to achieve the best prediction results.

Lastly, in Biomedical domain, having an exact answer is crucial and most of the research in this domain is focused on training with positive samples. On the other hand, when negative example is encountered while evaluation, the model attempts to predict an answer which is closest to the correct answer. As a result, the model's robustness shrinks despite of the certainty of correct answer prediction. Finetuning BERT on negative samples helps the prediction model to understand the no–answer possibility and to obtain unbiased results. We are interested in knowing the overall model performance upon the addition of negative samples to the dataset which teaches the model to distinguish between available and absent answer.

1.4 Thesis Motivation

Answering questions is a difficult Natural Language Processing task. Even though Deep Learning models enable the processing of massive amounts of data in a relatively short time; It is difficult to generalise model performance to the biomedical domain without similar data for finetuning. Pretraining the BERT model from scratch with biomedical literature exposes the model to biomedical vocabulary and context. Moreover, the model should be trained on handling question answers in the biomedical domain. As a result, the biomedical question-answering system needs to be improved by finetuning the BERT model on a large dataset, like SQuAD. All traditional methods focused on answerable examples without considering examples with no answers i.e., negative samples. To obtain unbiased results, it is critical to better represent data in the dataset while working on data augmentation. As a result, studying the model's performance when confronted with negative examples can aid in the accurate prediction of answers to biomedical question answering.

This research aims to propose a model that outputs the start and end index scores for each word in the context while predicting answers. The span selection technique is used to select the start and end index positions by evaluating all possible combinations of start and end scores, which is time-consuming. As a result, research to optimise the span selection technique that evaluates faster and improves prediction is required. In this thesis, we aim to provide enhanced datasets and techniques needed to improve the model's overall performance.

1.5 Thesis Contribution

This thesis work presents a new technique for span selection i.e., the start and end token position of an answer in the context. The technique discussed in section 4.2 primarily focuses on the extraction of the exact answer or no-answer prediction utilising less computation time and improving the results. When compared to the baseline model performance, improvements are observed in terms of both computation time and prediction accuracy. We also addressed the challenge of large biomedical dataset availability required for BERT finetuning. The problem is solved by increasing the size of the existing biomedical dataset. The presented data augmentation method creates multiple-question context pairs by dividing the existing context into segments in a way so that the sentence continuity and the meaning of each word in the sentence is maintained. According to our experiment, our data augmentation can be tuned to provide promising results.

Furthermore, we finetuned BERT on mixed dataset created by varying the ratio of positive and negative samples for better dataset representation. This helped us understand the model's behaviour towards unanswerable examples and learn the optimal number of negative samples to be added to attain maximum accuracy while predicting positive and negative samples.

1.6 Thesis Organization

The rest of the thesis is organized in the following manner:

In **Chapter II**, we present deep learning approaches such as recurrent neural networks (RNN), Long Short-Term Memory (LSTM), Embeddings from Language Models (Elmo), and Transformers for Natural Language Processing. We also present a Bidirectional Encoder Representation from Transformer (BERT) that can be used for a downstream task such as question answering.

In **Chapter III**, we present related work and literature that explains Question answering and its approaches, as well as biomedical question-answering techniques such as model training on various biomedical corpora for answer prediction. This section also discussed the state-of-the-art techniques used for biomedical question answering.

In **Chapter IV**, we discuss our proposed method for biomedical question-answer prediction. We discuss data augmentation and data sampling techniques to address the lack of large expert-annotated biomedical datasets availability. We also explain the proposed span selection technique for improving the model's overall performance.

In **Chapter V**, we explain the dataset used for performing the experiments along with the experimental setup with technical details of the platform the BERT setup. We provide information about the dataset used for training the model and performing the evaluations. We also present the results of the experiments conducted on the proposed approach and compare the same with the baseline methods. For evaluation and comparison of the performance, we used accuracy and F1 score.

With **Chapter VI**, we conclude the research highlighting what can be drawn from the results obtained and the scope of future work that can be taken up using the proposed approach.

Chapter 2

Background Study

This chapter describes the technical background of the study including the technologies related to this research. We discuss the most widely used deep learning approaches for natural language processing, such as recurrent neural networks (RNN), Long Short-Term Memory (LSTM), Embeddings from Language Models (Elmo), Transformers, and Bidirectional Encoder Representation from Transformer (BERT) model for question answering.

2.1 Feed Forward Neural Network (FNN)

An artificial neural network tries to replicate the neural connections in the human nervous system. Initially, neural networks were employed to tackle straightforward categorization issues, but as computing power has increased, more potent architectures are now available to handle more challenging cases. A feedforward neural network is one of them. Neural networks are used in solving problems like machine translation, search engines, mobile applications, and computer assistants.

Feedforward neural networks consist of the following:

Input Layer: Neurons at the input layer are responsible for receiving inputs and sending them to the other layers. The properties or features in the dataset represent the number of neurons in the input layer.

Output Layer: Depending on the model you are developing; the output layer is the anticipated feature.

Hidden Layer: Neurons in hidden layers alter inputs before passing on. As the network is trained, the weights are changed to be more predictive.

Neuron Weights: A neuron's weight is the level or quality of a connection between two neurons.

As shown in *Figure. 2.1*, a neural network is trained using data with the four features x1, x2, x3, and x4 and three neurons in the hidden layer. This neural network then performs a mathematical operation utilizing the weights (w) and bias (b) values. A bias value is included in the computation to improve the weighted sum of the inputs' accuracy in correctly fitting the model to the available data. An activation function helps to compute the neuron's output, resulting in a non-linear output. The output is computed using the below formula.

$$F(x) = \sum wixi + b \ (i=1 \text{ to } n)$$

i.e., n-total inputs to the neuron

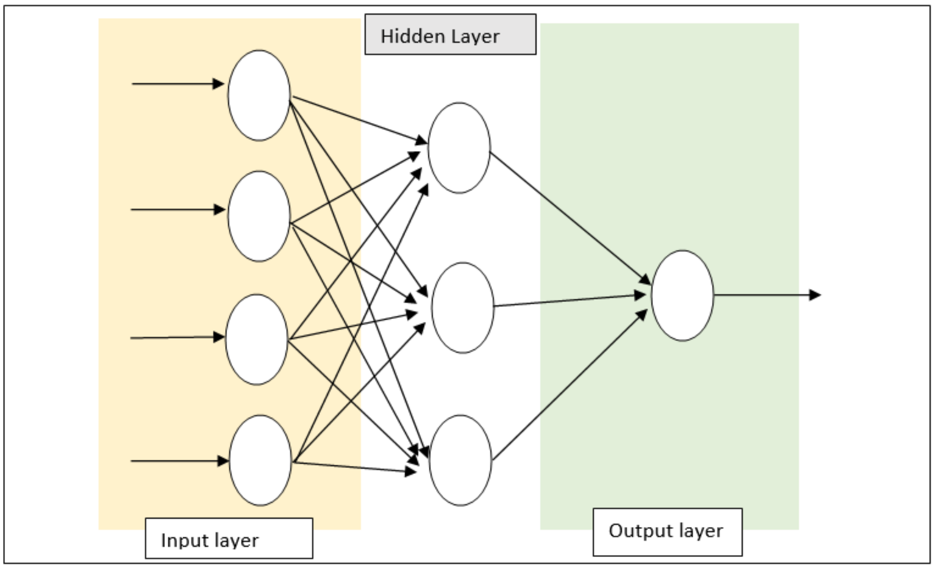


Figure 2.1 Feed-Forward Neural Network

2.2 Recurrent Neural Networks (RNN)

A recurrent Neural Network is a Feed-Forward neural network that helps sequential data. RNN algorithm produces predictive results in sequential data, which is Simple ordered data with related items following each other. The most usual form of sequential data is time series data, a collection of data points listed in chronological order. Unlike a typical Feed-forward network, while predicting the following word in the sequence, RNN considers both the current (x) and previous input(wh-1), which benefits the model to remember and process the historical information crucial in language modelling.

Different RNN types are proposed for various applications, such as

- One-to-one RNN used for general machine learning problem-solving
- One to many types for image captioning
- Many to one type apply to sentiment analysis
- Many to many types is employed for machine translation

RNN experiences the vanishing gradient problem, I.e., when a gradient's values are too small, this causes the model to stop learning or learn too slowly. Gradient calculates how much a function's output will vary if its inputs are slightly altered.

RNN has a short-term memory; therefore, it captures only short-term dependency, thus making it difficult to predict the following output precisely.

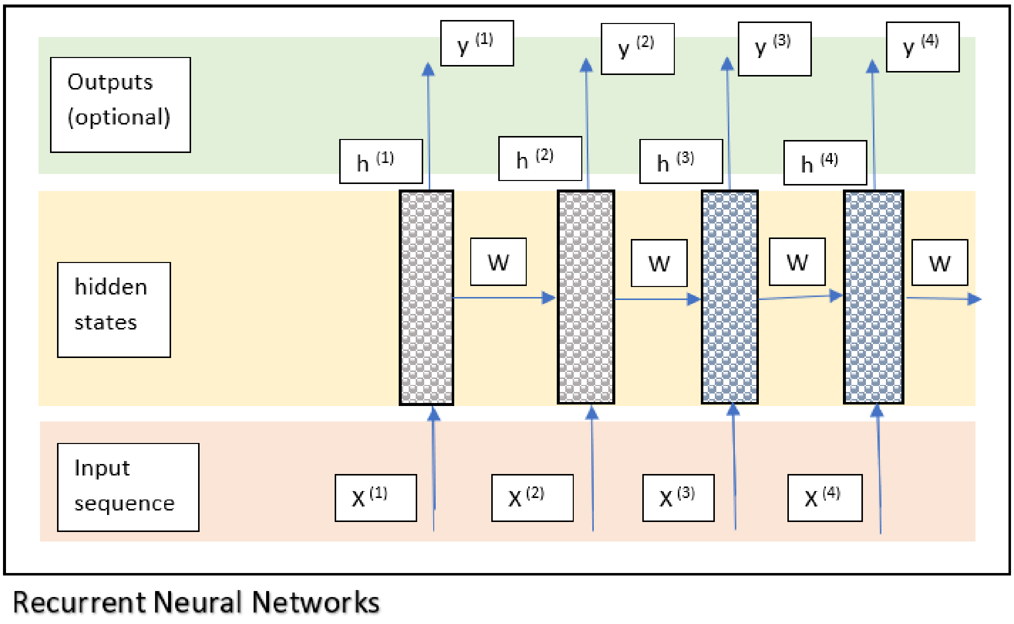


Figure 2.2 Recurrent Neural Networks

2.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory is an extension of RNNs that overcomes the problem of long-term dependency. LSTMs can retain inputs for a long time because it has a memory that stores information for extended periods. The LSTM can read, write, and delete data from its memory. The memory resembles a gated cell where the cell determines, based on the value it assigns to the information, whether to store or erase information. It helps the algorithm to learn what information is crucial for next-term prediction.

LSTM architecture consists of input, forget, and output gates. These gates decide whether to allow additional input (input gate), erase the data because it is unimportant (forget gate), or enable the information to affect the output at the current timestep (output gate).

LSTM neural networks perform various tasks such as Machine translation, Question Answering, Language Modelling, and Image Captioning. Although the model resolved the vanishing gradient problem, it did not remove it altogether. Moreover, as shown in the *Figure 2.3*, it is a computationally complex solution that requires ample time to train and become application ready.

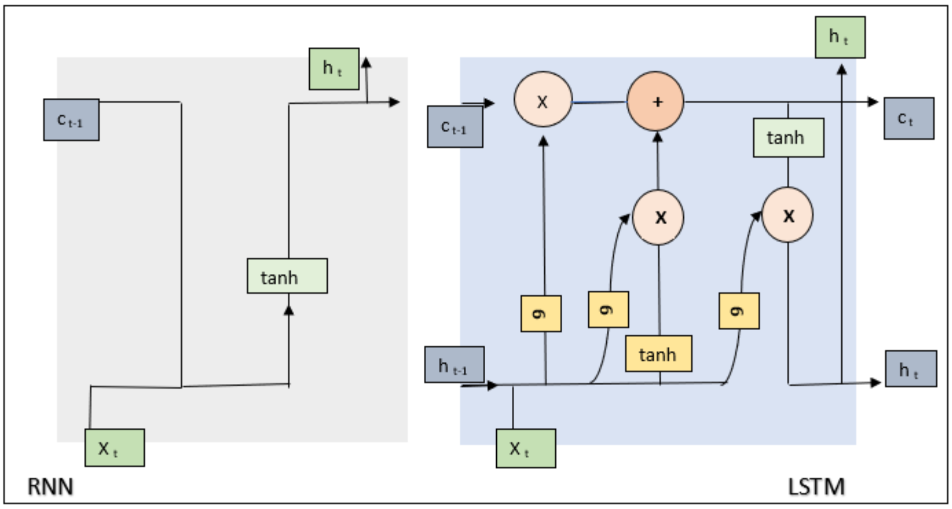


Figure 2.3 Long-Short Term Memory

2.4 Embeddings from Language Models (Elmo)

A novel solution for word embedding representations Elmo computed on top of a two-layer bidirectional language model (bi-LSTM). This model has two layers, each consisting of a forward and backward pass. As *Figure 2.4* shows, the raw word vectors are given as inputs to the first layer of bi-LSTM. The forward pass corresponds to a specific term and the context (other words) before that word, Whereas the backward pass stores information about the word and the context after it. The intermediate word vectors obtained from both passes are fed as input to the next layer. The resultant embedding is the weighted sum of the raw word vectors and the two intermediate word vectors.

Elmo embeddings are context-dependent, which means that depending on the context in which a word is used, these models supply various vector representations (embeddings) for that word. The model uses the character-level convolutional neural network to capture the inner structure of the word.

Consider these two sentences.

1. Every stick of furniture just vanished

2. Stick the balls of wool on knitting needles.

The first sentence stick is a noun, and the second is a verb. Depending on the sentence in which it appears, each of these words has a distinct meaning or interpretation. Previously, word embeddings produced the same vector for the word "stick" in both phrases without considering the usage context. Thus, Elmo successfully overcomes this limitation by calculating the word vector based on the context of the sentence.

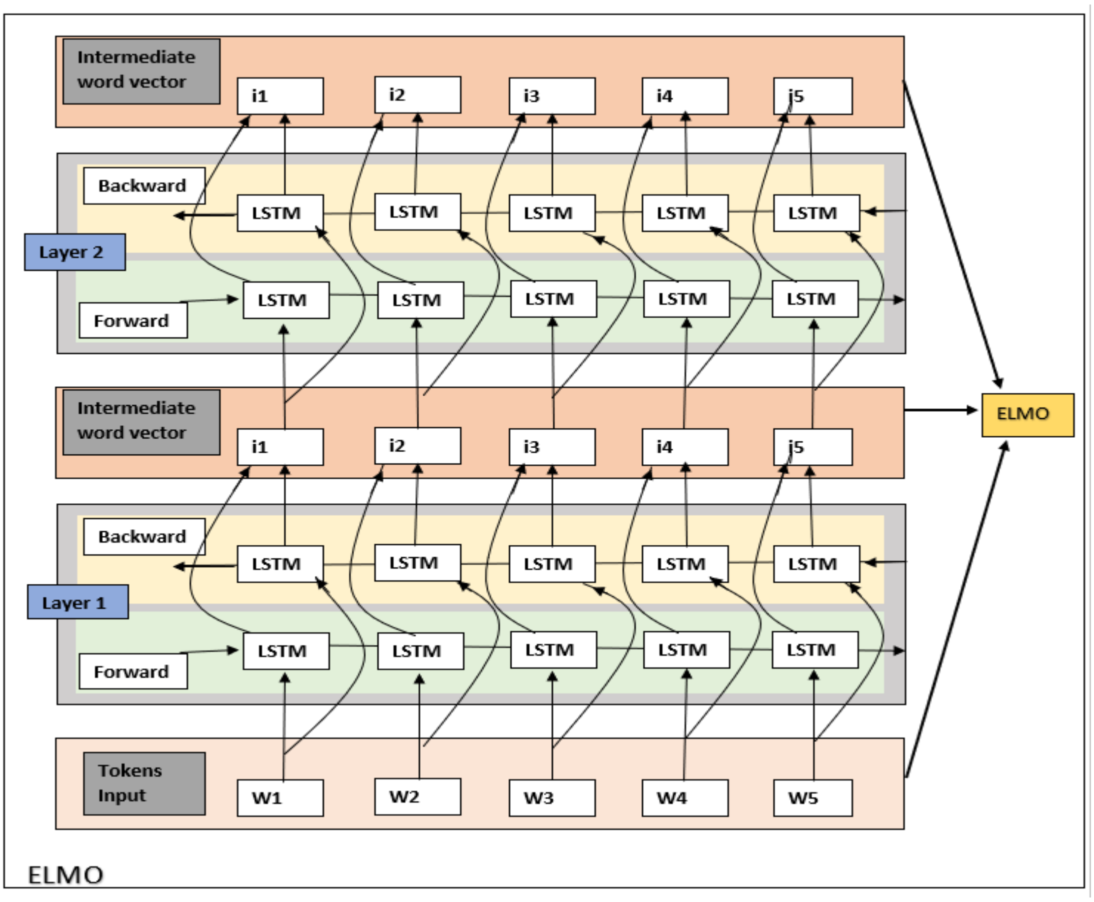


Figure 2.4 Embeddings from Language Models

2.5 Transformers

A novel neural network based on a self-attention mechanism is well suited for language understanding. The transformer is an encoder-decoder architecture where the same units of encoder-decoder are stacked on top of each other.

Let us understand the transformer with an example: Task -Language Translation

Input: English

Diarrhea can cause stomach infection

Output: French

La diarrhée peut causer une infection de l'estomac

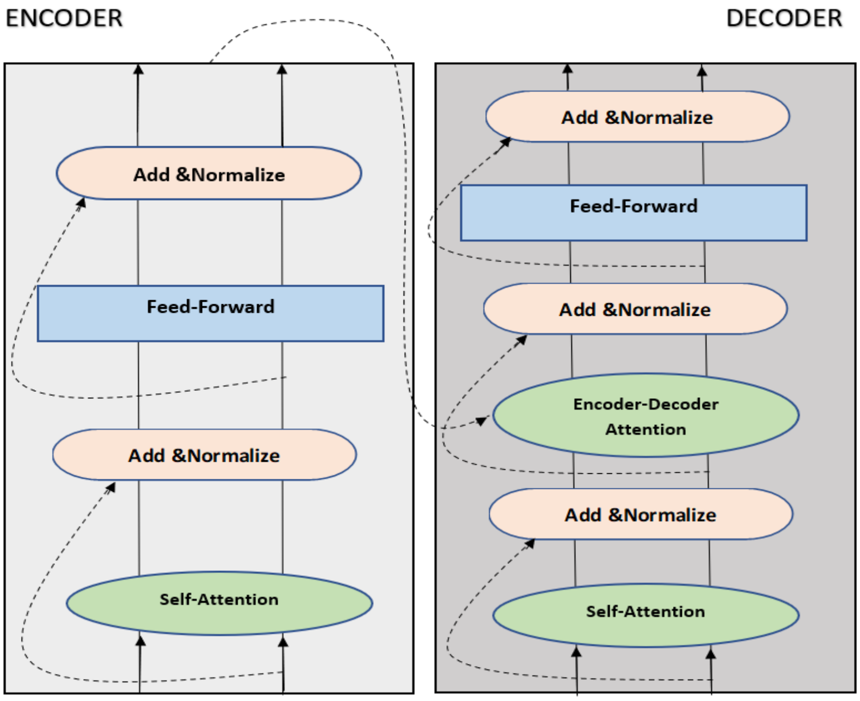


Figure 2.5 Transformer Architecture

2.5.1 Encoder

The encoder sends word vectors through a self-attention layer, feed-forward neural network layer, and normalize layer responsible for normalizing each layer's output to make it suitable for the next layer's input, finally sending the result upward to the succeeding encoders. As shown in *Figure 2.5*, the dotted line shows that the pre-processed input is fed into the layer to retain any information lost by the layer directly below them.

Encoder Input: *Figure 2.6* represents the input sequence where each word is tokenized as a word vector, and positional encodings are applied to word vectors to preserve the order of the words in the sequence. The word vectors are inputs to the bottommost encoder. All other encoders receive input from the encoder directly below them. Input word embeddings consist of a vector of size 512, as shown in *Figure 2.7*.

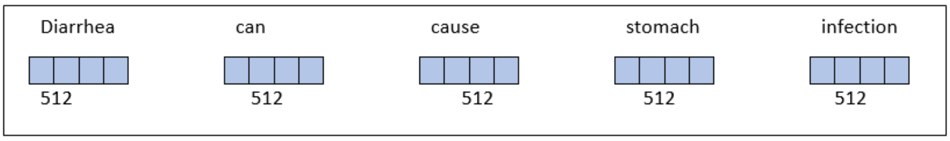


Figure 2.6 Embedding for Input sequence

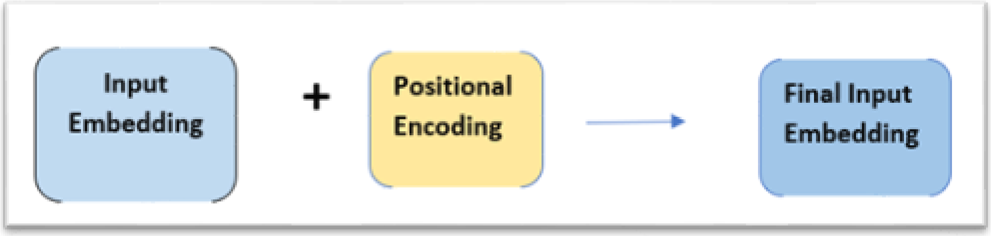


Figure 2.7 Final input fed to an encoder

Self-Attention: Self-attention scans the relationship of each word in the input sequence with its adjacent words for hints that may improve the encoding of the target word. For example, let us consider the target word, Diarrhea. The attention layer calculates the relevance of Diarrhea with other words in the sentence, thus generating an attention vector for Diarrhea. *Figure 2.8* shows different shades of colour based on their contribution to the sequence concerning the target word 'Diarrhea.'

An attention vector is generated for each target word, and its values correspond to the adjacent word's contribution to the target word in the sentence. For example: Diarrhea [.51 .06 .30 .10 .14] I.e., Diarrhea (51%), can (6%), cause (30%), stomach (10%), infection (13%) total 100% contribution towards the entire sentence with respect to the target word 'Diarrhea'.

In the design of the Transformer, self-attention is computed multiple times, independently and concurrently. As a result, it is known as multi-head attention.

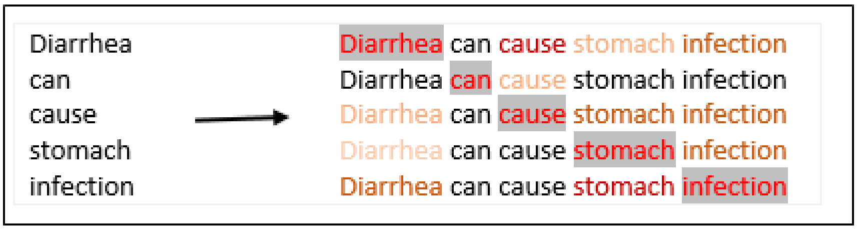

Diarrhea
can
cause
stomach
infection
Diarrhea can cause stomach infection
Diarrhea can cause stomach infection
Diarrhea can cause stomach infection
Diarrhea can cause stomach infection
Diarrhea can cause stomach infection

Figure 2.8 Self-Attention calculation of each word in the sentence

Feed Forward Neural Network: Attention vectors representing each input word are subjected to the Feed Forward network to convert it into a readable form for processing by the next block. These attention word vectors are independent of each other; thus, parallelization is used to formulate the output, as shown in *Figure 2.9*.

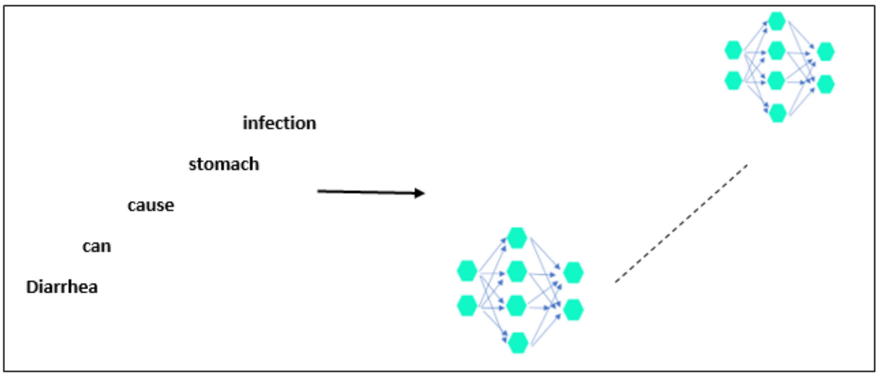


Figure 2.9 Input words (attention vectors) fed into FFN parallelly

Layer Normalization: It is responsible for making each layer's output suitable for the input of the following layer before sending the result upward to the subsequent encoders. Thus, making the intermediate outputs word vector suitable as input in the following layer.

2.5.2 Decoder

Like the encoder, the Decoder sends word vectors through a self-attention layer, feed-forward neural network layer, normalization layer and an additional encoder-decoder attention layer which is crucial for the transformation. The dotted line in *Figure 2.5* shows that pre-processed input is sent into the layer as depicted to retain any information lost by the layer directly below them.

Decoder input: Positional encoding is applied to the decoder input, which is the predicted words in the previous iteration cycle of the decoder.

Self-attention: Generates the Attention vector of the target French word and its relevance with the predicted words in earlier iterations, I.e., the Contribution of the last word to the target word in the sentence.

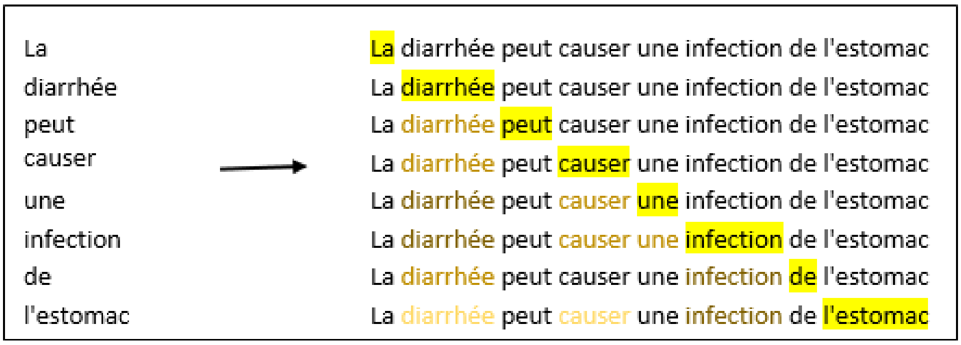

La
diarrhée
peut
causer
une
infection
de
l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac
La diarrhée peut causer une infection de l'estomac

Figure 2.10 Decoder Self-Attention layer

Encoder-Decoder Attention Block: The encoder-decoder attention is trained to associate the input sentence with the corresponding output word. The relationship between each English word and each French word will eventually be determined. This is where the mapping between English and French takes place.

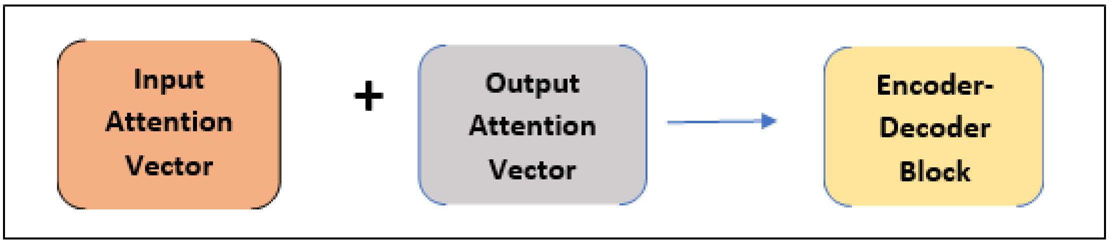


Figure 2.11 Encoder-Decoder Attention Block

The output vectors of the decoder blocks are then fed into Linear Layer (Feed-Forward network layer) which expands the number of dimensions into French language. SoftMax layer forms Probability Distribution which can be easily interpreted. Thus, we get a translated word with every iteration until end of the sentence is reached.

2.6 Bidirectional Encoder Representations from Transformer (BERT)

BERT is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus [7]. Without needing to significantly alter model architecture for each unique purpose, the pre-trained BERT representations can be fine-tuned with just one additional output layer to produce models for a variety of activities such as Question Answering.

Training BERT on two tasks:

1) Masked Language Model: Masking out some of the words in the input and then conditioning each word bidirectionally to predict the masked words

Input: Acrokeratosis paraneoplastica [MASK1] syndrome associated with esophageal [MASK2] carcinoma.

Labels: [MASK1] = Bazex , [MASK2] = squamocellular

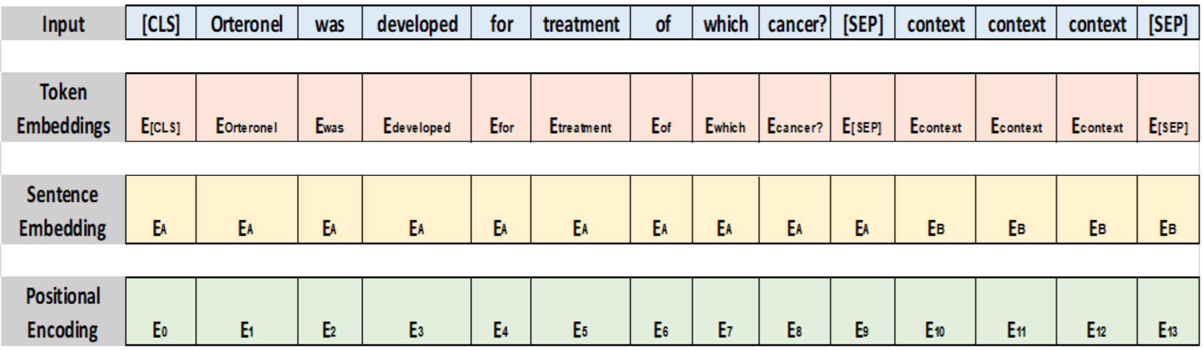
2) Next Sentence Representation: To model relationships between sentences by pre-training on a quite simple task that can be generated from any text corpus

Sentence 1: The models were pre-trained within an encoder decoder framework.

Sentence 2: These works showed the potential relevance of neural networks and traditional machine learning methods in the detection of question similarity and entailment.

Label = NextSentence

Input of the Bert consists of sum of three embeddings namely Token Embedding (in this case used 30,000-word piece vocabulary), Segment Embedding and Position Embedding



Input	[CLS]	Orteronel	was	developed	for	treatment	of	which	cancer?	[SEP]	context	context	context	[SEP]
Token Embeddings	$E_{[CLS]}$	$E_{Orteronel}$	E_{was}	$E_{developed}$	E_{for}	$E_{treatment}$	E_{of}	E_{which}	$E_{cancer?}$	$E_{[SEP]}$	$E_{context}$	$E_{context}$	$E_{context}$	$E_{[SEP]}$
Sentence Embedding	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B
Positional Encoding	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}

Figure 2.12 BERT Input Representation

Finetuning: Question Answering Task

To fine-tune the Bert model for a QA assignment, a question context pair is provided as input, and the output is the index position of the response within the context. The weights are modified based on the accuracy of the answer position so that in subsequent iterations, the Bert model would predict answers that were more closely aligned with the actual answers. The model can be fine-tuned such that it can recognize the proper answer to a specific question given the question context pair.

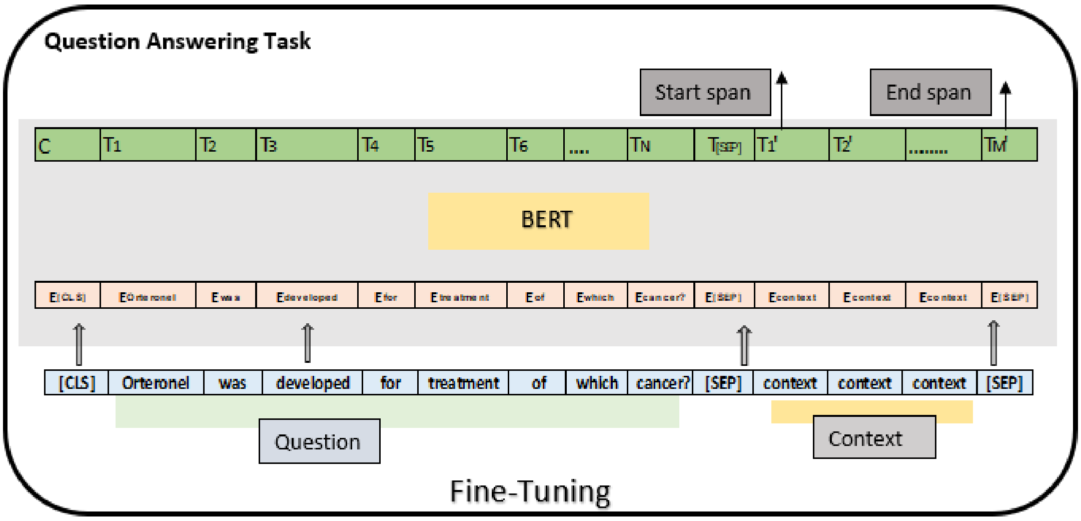


Figure 2.13 Question answering Architecture

Chapter 3

Related Work

This chapter presents a summary of the research done in the biomedical field to use deep learning methods like transformers to address the issues with biomedical question-answering. It also presents the traditional question answering approaches like rule-based, Statistical and Machine learning. Then, we go through Biomedical question answering approaches and deep learning techniques. We examine the important works of related literature to comprehend the biomedical question-answering process and the application of BERT to this problem.

3.1 Question Answering

Biomedical professionals and the public need practical assistance to access, understand, and consume complex biomedical concepts. For instance, doctors must be aware of the most recent clinical evidence for diagnosing and treating diseases under the Evidence-Based Medicine framework, and the public is becoming more interested in learning their health information. The recent COVID-19 lockout has reminded us how crucial a question-answering system, particularly one that deals with health, is when the public needs trustworthy information at the convenience of their homes. More healthcare services have emerged due to recent advancements and the growth of big data, including online medical information retrieval and biomedical question-and-answer applications that can assist people in finding health information and biomedical knowledge quickly and affordably. Biomedical question answering technology, a sub-task of natural language processing in the biomedical domain that can discover and extract necessary biomedical text spans, is a fundamental and practical way for knowledge retrieval and representation in various healthcare application situations.

The problem of question answering (QA) in natural language understanding has proven difficult. The essential element in QA demands the capacity to comprehend the query and the setting in which the query was conceived. QA has been deemed difficult because of the changing nature of the languages [21]. One of the most popular QA systems approaches was the rule-based approach [22]. These systems used principles derived from grammatical semantics for each question to select the correct response. These guidelines are typically heuristic and handmade, based on lexical and semantic cues from the surrounding environment [22]. Adding syntactic analysis, morphological analysis, Part-Of-Speech tagging and Named Entity Recognition improved the answer matching of these systems [22].

However, [23] point out that that manually built heuristic rules exploit predetermined patterns that group questions according to the type of answer. Thus, rules with a thorough understanding of the Language semantics were crucial.

Statistical approaches deal with a substantial amount of data and diversity. It is independent of structured query languages and can formulate queries in natural language form. So far, statistical techniques have been successfully used at several levels of a QA system like question analysis, document analysis and answer analysis [24]. Support vector machine (SVM) classifiers, Bayesian classifiers, and Maximum entropy models are some techniques used for question classification. The system was able to produce a training set after analysing an annotated corpus of questions or articles with topics, like biomedical, thanks to statistical processes. Thus, it became possible for machine learning to analyse annotated corpora (training set) and subsequently create knowledge bases. Combined with statistical approaches, machine learning has developed Successful systems in linguistic and sentiment-related fields [25]. If there is enough training data, named entity recognition algorithms process the context, operate as the classifier, and act as a knowledge base, which helps the learning potential of machine learning system be extremely scalable [6].

Deep learning approach learns underlying features in data using a Neural network, a network of connected neurons producing a sequence of real value activations. Neural network designs utilized logical representations from the text's context to predict the answer. Convolutional Neural Networks appeared to be a game-changing technique in various fields, including text, video, and speech, but they could not simulate lengthy dependencies, which are crucial for language processing. Recurrent neural networks (RNNs) are better at processing sequential data because they can operate on an input sequence depending on the outcomes of earlier computations. RNNs can memorize the previously fed inputs and produce an output based on them. Thus, making RNNs adaptable and valuable for time-series data, text, audio, and other sequential data types [26]. This RNN characteristic enables the network to compare group histories, effectively representing patterns with varying lengths. However, recurrent networks do have some drawbacks. The main problem with recurrent neural networks is that gradient computation worsens when an error signal is propagated back in time, making it impossible to learn phenomena with arbitrarily long periods [27]. Long Short-Term Memory (LSTM) architectures enhanced performance in areas like text classification and machine translation [28] by overcoming short dependency problems faced with RNNs architecture.

Elmo exhibits improvements in various NLP tasks as a general method for learning high-quality deep context-dependent representations from biLMs. Elmo effectively applies biLM layers to different syntactic and semantic information about words in context, increasing task performance overall [29]. With the introduction of a multi-layer, compared to recurrent neural networks, such as LSTM, the multi-head self-attention mechanism has shown superiority in exploiting GPU-based parallel computation and modelling long-range dependencies in texts. The transformer focuses on the global dependency between input and output based on attention mechanisms. It follows parallelization and achieves state-of-the-art results for language translation [3]. BERT: Bidirectional Encoder Representations from Transformers, a pre-trained model to efficiently finetune downstream tasks, have gained tremendous momentum. Pretrained on two tasks (i) masked language model that enhances the objective is to predict the original vocabulary word based only on its context and (ii) next sentence prediction to have a better grasp on language understanding [9].

3.2 Question Answering Approaches

Knowledge acquisition is a crucial activity in knowledge management and information retrieval. Answering questions (QA) Systems that directly respond to user questions are simpler and more intuitive. QA is a challenging natural language processing (NLP) benchmarking task to present an intelligent system that analyses questions, finds and uses appropriate materials, and generates answers [1]. Studies categorize Question Answering into three modules: Question Processing, Document Processing, and Answer Processing.

Question Processing: It focuses on evaluating and classifying questions to discover the question type and answer expectations to avoid any uncertainties while answering the question [31]. It converts questions into search queries based on the essential word weightage. Deep learning technology contributes to determining the context of each word in the sentence. Research [32] highlights the importance of converting semantic relations into machine-readable questions to analyse natural language questions efficiently. In order to produce correct results, questions can be categorized based on the predicted answer type by adhering to specific defined rules and using manual and automatic approaches, but the processes were tedious and time-consuming. Moreover, questions are described based on the answer type obtained. The question types include general questions with Yes/No answers, factoid questions, summary questions and list questions [33]. For questions with a Yes/No answer, one of two options is the anticipated response; one answers the question affirmatively, and the other denies it [34]. Typical factoid interrogations involve asking a question regarding a straightforward fact and obtaining a brief response. A list question requires a set of items that meet the specified requirements. Summary answers are generated from scratch that is neither yes/no nor factual ones extracted from the contexts.

Document Processing: The main aim is to choose a collection of relevant documents, texts and extracts depending on the question's focus using NLP. Based on its rank for relevance to the given question, the collected material is stored [35]. Top-ranked documents to relevant questions are split into multiple sentences and matched for exact words in the question. The sentence containing the most similar words is chosen as a highly classified sentence.

Answer Processing: This module focuses on combining data from several sources, summarising information, removing contradictions or uncertainties, and extracting the answers to the targeted question from a related document [36]. The answer is discovered from the labelled corpus by analysing the question words and the expected answer labels that go with them. Machine Learning and NLP methods like probabilistic, algebraic, and neural networks contribute to solving various answer-processing issues [37].

3.3 Biomedical Question Answering techniques

A domain-specific language representation model called Bio BERT has already undergone extensive training using sizable biomedical corpora. [8] Extracts from papers in PMC and PubMed. It uses Word Piece tokenization, where any new words can be represented by commonly occurring subwords. Immunoglobulin, for instance, becomes I ##mm ##uno ##g ##lo ##bul ##in. BioBert was finetuned for the task of answering questions using the BioASQ 4b-, 5b-, and 6b-factoid data sets. Strict accuracy, lenient accuracy, and mean reciprocal rank were the evaluation metrics employed.

The recognised differences in linguistic features between clinical narratives (such doctor notes) and both general text and non-clinical biomedical literature are what led to the need for specialised clinical BERT models. [10] It pre-trained on the MIMIC-III v1.4 dataset and used the same word piece tokenization as BERT. Named Entity Recognition (MedNLI) and Natural Language Inference (i2b2) are two tasks for which the clinical BERT is optimised. It is assessed on Accuracy and Exact F1 metrics.

To enhance performance on upcoming scientific NLP tasks, SCI BERT makes use of unsupervised pretraining on a sizable multi-domain corpus of scientific literature [11]. Utilizing the SentencePiece library, WordPiece (SCIVOCAB) - 30K was created. Base vocab and Sci vocab overlap 42 percent of the 1.14 million biomedical (82 percent) and computer science (18 percent) documents from the Semantic Scholar corpus. For several tasks, including as Text Classification, Relation Classification, Dependency Parsing, and Named Entity Recognition, macro F1 metrics were used.

To aid in the investigation of pre-training language representations in the biomedical domain, (BLUE) benchmark has been developed [12]. The model is Pre-trained on PubMed abstracts and MIMIC-III clinical notes. They optimised four BERT models, including BERT-Large (P+M), BERT-Base (P+M), and BERT-Large (P), for a variety of tasks, including Named Entity Recognition (DDI, ChemProt, i2b2 2010), Relation extraction, and Sentence Similarity (BioSSES and MedSTS) The metrics assessed included accuracy, micro-F1 and Pearson.

UmlsBERT, a contextual embedding model that incorporates subject-matter expertise before training [13], By utilising data from the semantic type of each (biomedical) word, it generates input embeddings that are more meaningful, connecting words in UMLS WordPiece tokenization that have the same fundamental idea as that base BERT model.

In [14] BERT is trained to deduce the associated disease and aspect from a disease description text using weakly-supervised signals from Wikipedia. It utilises disease knowledge from wikis and WordPiece tokenization when pre-training the BERT model. Tasks have been fine-tuned Question Answering - (MEDIQA-2019, TRECQA-2017) Named Entity Recognition (BC5CDR-disease, NCBI) and Inference disease, (MEDNLI).

Pretraining in a particular domain from scratch can perform noticeably better than pretraining in multiple domains, such as pretraining continuously from a general-domain language model. Biomedical Language Understanding and Reasoning Benchmark (BLURB)[15] uses biomedical vocabulary and achieved better results than state of the art Bert models. The model has been enhanced on a variety of tasks, including question answering, sentence similarity, and document classification.

Question answering (QA) systems have recently delivered improvements, but generalizing the model is challenging. Training QA models with various question context pairs via question paraphrase may increase their robustness. The intention is to confuse the model and get it to produce the wrong responses by changing the question's semantics. Previous work done in question paraphrasing shows that QA models can be confused by adding distracting sentences at the end of the context, which is the opposite of the existing natural context [38]. Question paraphrases are generated by determining the most critical question word and swapping it with a synonym taken from WordNet and Elmo embeddings and approved by human annotators. Even though the original questions' sense is preserved, initial correct responses may occasionally alter [39]. In addition to generating a single paraphrase, the benefit of generating several paraphrases for a particular input sentence has also been investigated.

Multi-domain training enhances the generalization and robustness of models and emphasizes the need for diverse question-answering domains [40]. By adding a, No Answer option to the usual phrasing of the problem, datasets like SQuAD 2.0 have extended extractive question answering. [41] has shown that creating context paraphrases using back translation significantly boosts reading comprehension on the complex SQuAD 1.1 assessment. The selection of No Answer segments yielded the most significant gains in multi-domain generalization and was especially useful for extractive model training on large sequences [2].

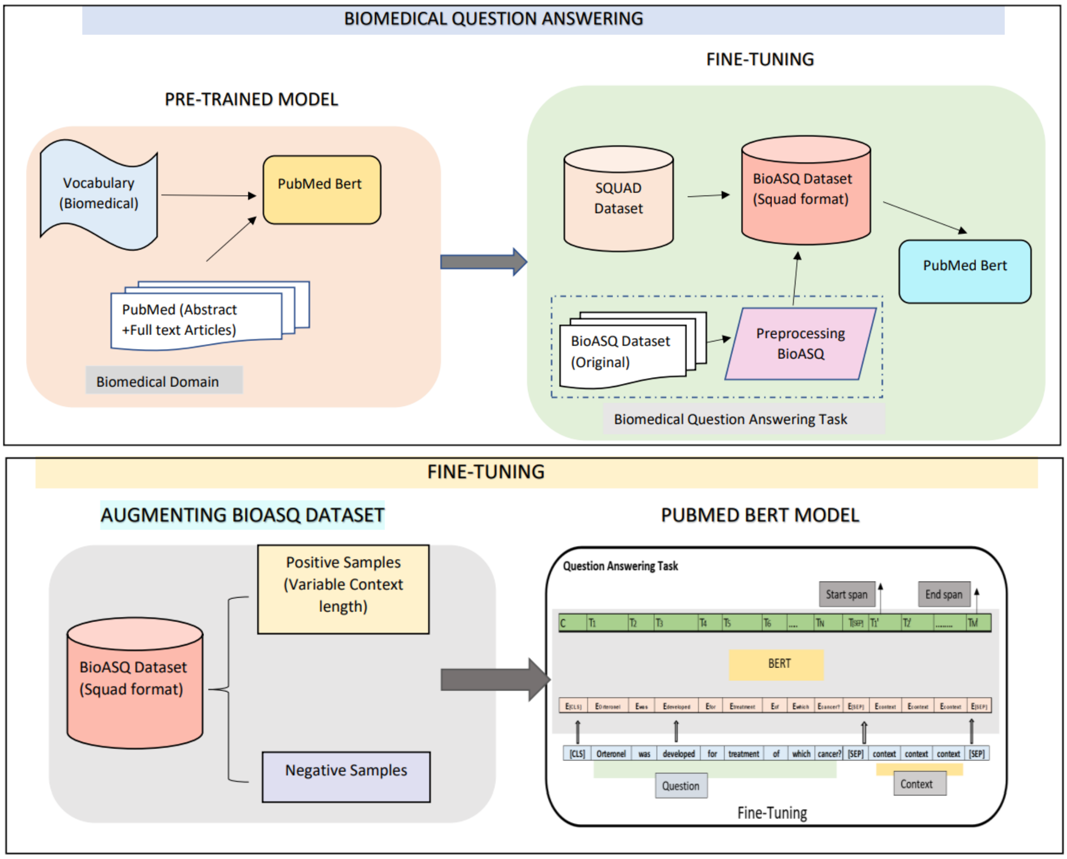
Models	Vocab	Pretraining	Finetuning	Metrics
BERT [9]	Wordpiece (General)	wiki+Books	(QA) - SQuAD v1.1,SQuAD v2.0	GLUE, F1, EM
BioBERT [8]	Wordpiece (General)	PMC+Pubmed	(QA) - BioASQ Factoid 4b,5b,6b	strict accuracy, lenient accuracy and mean reciprocal rank.
Clinical Bert [10]	Wordpiece (General)	MIMIC-III v1.4	(NLI) - MedNLI, (NER) - I2b2 06,10,12,14	Accuracy, exact F1
SciBert [11]	Wordpiece (SCIVOCAB)	Semantic Scholar corpus - biomedical (82%) and CS papers (18%)	(NER) - BC5CDR, JNLPBA, NCBI-disease (REL) - ChemProt	macro F1
BLUE [12]	Wordpiece (General)	wiki+books+pubmed+mimic	(NER) - BC5CDR, ShARe/CLEFE (RE) - DDI, ChemProt, i2b2 2010 (Sentence similarity) - MedSTS, BIOSSES	Pearson, F1, micro F1,accuracy
UmlsBert [13]	Wordpiece (General)	Bio_ClinicalBERT model + MIMIC-III	(NER) - i2b2 (NLI) - MedNLi	Accuracy, F1
BERT + disease [14]	Wordpiece (General)	model+disease knowledge infusion (wiki)	(QA) - MEDIQA-2019, TRECQA-2017 (NER) - BC5CDR-disease, NCBI (NLI) - MEDNLI	Accuracy, F1,
PubMed Bert [15]	PubMed vocabulary	PubMed abstracts	(QA) - BioASQ, PubMedQA	Accuracy, F1

Table 1: BQA techniques

Chapter 4

Proposed Approach

Natural Language Processing Task Question Answering has seen great success in the General domain since the introduction of deep learning model Transformer, specifically BERT. Efforts to replicate this success in the biomedical field are ongoing. The availability of a large manually annotated biomedical dataset is one of the most difficult challenges in biomedical question answering. As a result, this thesis proposes methods for augmenting the dataset based on context length and sample type, i.e., positive, and negative examples, detecting patterns in the results, studying the samples, and implementing a new technique to improve span selection problem of start/end token while predicting the correct answer to the given question context pair.


BIOMEDICAL QUESTION ANSWERING
PRE-TRAINED MODEL
Vocabulary (Biomedical)
PubMed Bert
PubMed (Abstract +Full text Articles)
Biomedical Domain
FINE-TUNING
SQUAD Dataset
BioASQ Dataset (Squad format)
PubMed Bert
BioASQ Dataset (Original)
Preprocessing BioASQ
Biomedical Question Answering Task
FINE-TUNING
AUGMENTING BIOASQ DATASET
BioASQ Dataset (Squad format)
Positive Samples (Variable Context length)
Negative Samples
PUBMED BERT MODEL
Question Answering Task
Start span
End span
C T1 T2 T3 T4 T5 T6 .... TN T[SEP] T1' T2' .... TM
BERT
E[CLS] EOrteronel Ewas Edeveloped Efor Etreatment Eof Ewhich Ecancer? E[SEP] Econtext Econtext Econtext E[SEP]
[CLS] Orteronel was developed for treatment of which cancer? [SEP] context context context [SEP]
Question
Context
Fine-Tuning

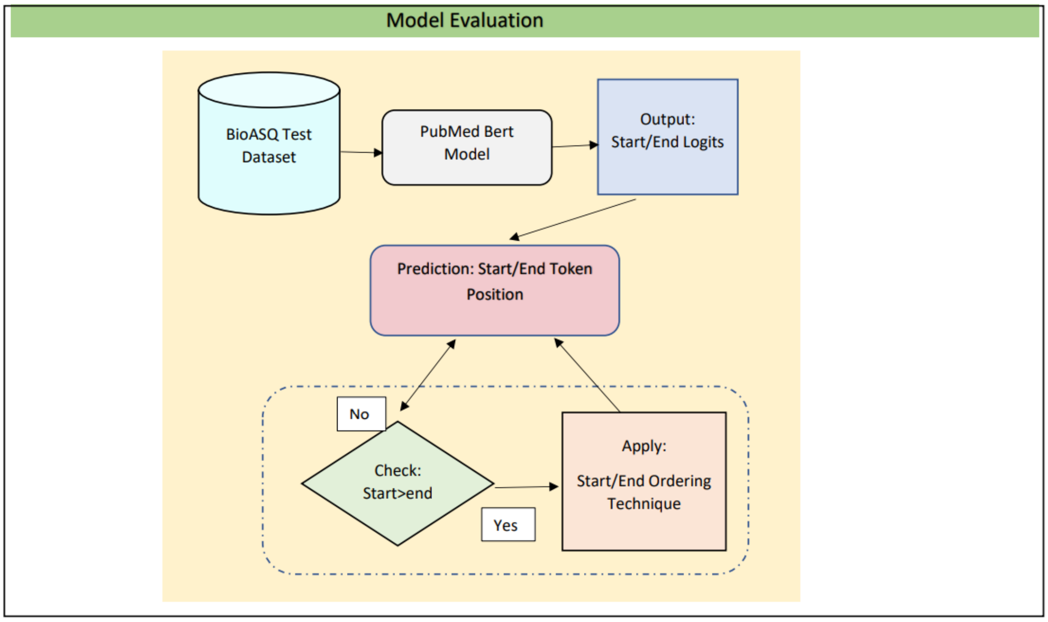


Figure 4.1 Architectural Diagram for Biomedical Question Answering

4.1 Question Answering Task

We focus on extractive question-answering in the Biomedical Question Answering Task, which involves extracting/highlighting the answer in the passage that answers the question. The Question Answering task entails feeding the model a question context pair as input and anticipating the start and end position of the answer within the context if the answer is present. Otherwise, it returns an empty response with start/end position as 0.

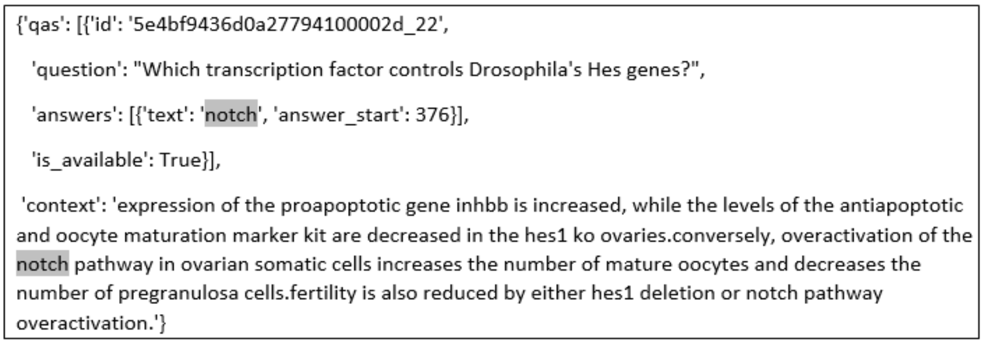

{'qas': [{'id': '5e4bf9436d0a27794100002d_22',
  'question': "Which transcription factor controls Drosophila's Hes genes?",
  'answers': [{'text': 'notch', 'answer_start': 376}],
  'is_available': True}],
 'context': 'expression of the proapoptotic gene inhbb is increased, while the levels of the antiapoptotic and oocyte maturation marker kit are decreased in the hes1 ko ovaries.conversely, overactivation of the notch pathway in ovarian somatic cells increases the number of mature oocytes and decreases the number of pregranulosa cells.fertility is also reduced by either hes1 deletion or notch pathway overactivation.'}

Figure 4.2 Biomedical question answering dataset example

Dataset: We chose BioASQ, the largest manually annotated dataset in the biomedical field. The dataset comprises a question (qas) and a context field, which is an abstract of the pertinent documents extracted to address the question. Additional items include the example unique id, the question text, the answer text, the answer start position and the Boolean value confirming answer availability in the context.

Model – The present thesis employs the PubMed Bert model, which is trained on PubMed abstracts and full-text articles from PubMed Central. To better satisfy the requirements of the biomedical area, PubMed Bert uses biomedical vocabulary. The model receives a question and context pair as input, and the maximum number of tokens it can process is 512 for each example.

The followings are the steps to be performed for Finetuning Question Answering task

1. Splitting data: The dataset is structured into several passages (or contexts) for each question and answer. There is a list of contexts, questions, and answers. When there are multiple contexts for each question, the questions are repeated.

As shown in the *Figure 4.2* each example is a dictionary containing view of key-value pair

(i) key: ’qas’ value: *id*-a number to uniquely identify the example,

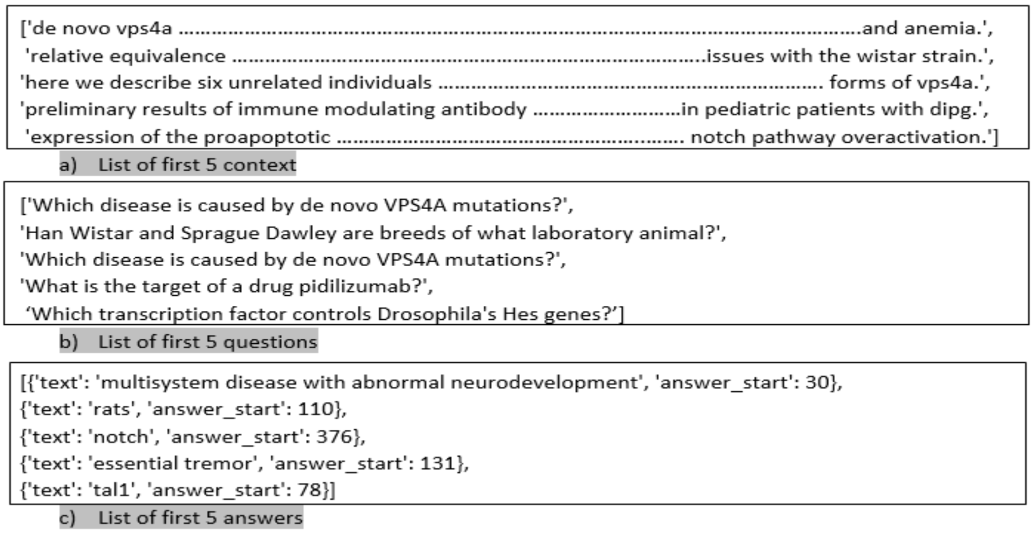
question- the actual question in text

answer- the answer in text

answer start position (character position),

Is_available- Boolean value to determine answer availability

(ii) key: ‘context’ value- *passage in text*



```
['de novo vps4a ……………………………………………………………………………………………………………….and anemia.',
 'relative equivalence ……………………………………………………………………………….issues with the wistar strain.',
'here we describe six unrelated individuals ………………………………………………………….. forms of vps4a.',
'preliminary results of immune modulating antibody ……………………….in pediatric patients with dipg.',
 'expression of the proapoptotic …………………………………………………….. notch pathway overactivation.']
```

a) List of first 5 context

```
['Which disease is caused by de novo VPS4A mutations?',
'Han Wistar and Sprague Dawley are breeds of what laboratory animal?',
'Which disease is caused by de novo VPS4A mutations?',
'What is the target of a drug pidilizumab?',
 ‘Which transcription factor controls Drosophila's Hes genes?’]
```

b) List of first 5 questions

```
[{'text': 'multisystem disease with abnormal neurodevelopment', 'answer_start': 30},
{'text': 'rats', 'answer_start': 110},
{'text': 'notch', 'answer_start': 376},
{'text': 'essential tremor', 'answer_start': 131},
{'text': 'tal1', 'answer_start': 78}]
```

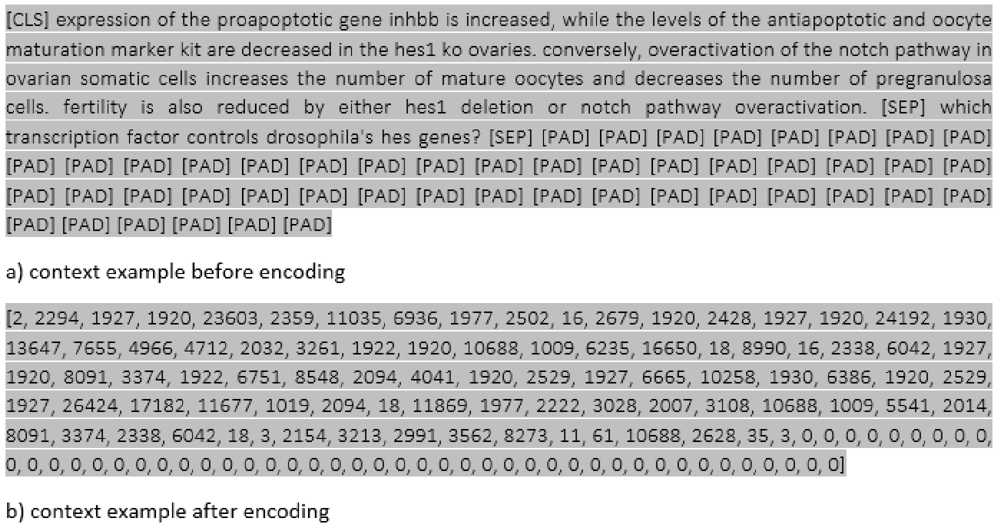
c) List of first 5 answers

Figure 4.3 List of questions, context and answers

The context and questions lists are just strings, and the answers list is a dictionary containing the context's substring with the correct answer text and an integer indicating where the character of the answer begins.

2. End character position: First, the context passage is used to determine the end character position of the answer based on the start character position and length of the answer. Adjustments are made to accommodate the situation when the end character's position is off by one or two positions.

3. Tokenization: The tokenizer converts each word of the question text and context into a token, which is the acceptable format for a Bert-like model (see *Figure 4.4* for an example of a conversion). Tokenizers take multiple lists of sequences (questions, context, and answers) and encode them into pairs of sequences. To train a model, one needs (i) tokenized context/question pairs and (ii) an integer of start/end token positions.



[CLS] expression of the proapoptotic gene inhbb is increased, while the levels of the antiapoptotic and oocyte maturation marker kit are decreased in the hes1 ko ovaries. conversely, overactivation of the notch pathway in ovarian somatic cells increases the number of mature oocytes and decreases the number of pregranulosa cells. fertility is also reduced by either hes1 deletion or notch pathway overactivation. [SEP] which transcription factor controls drosophila's hes genes? [SEP] [PAD]

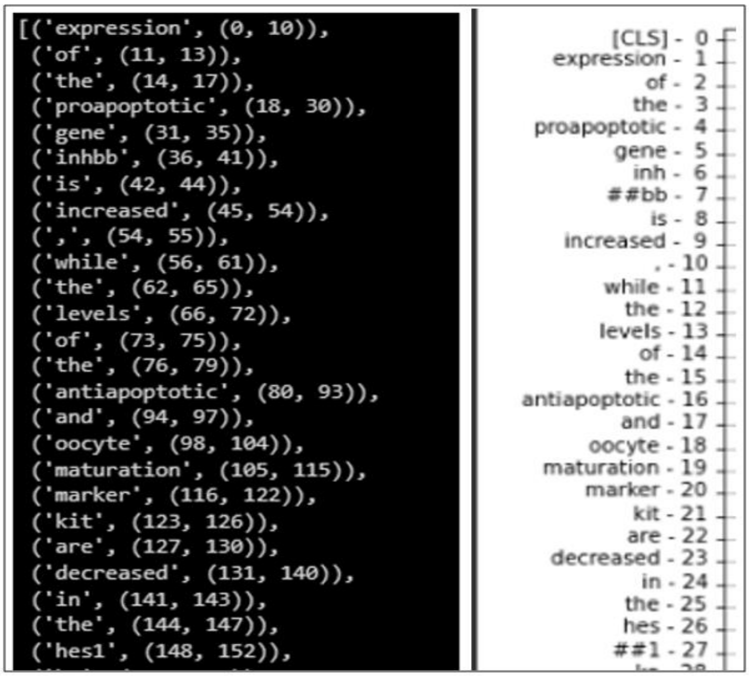
a) context example before encoding

[2, 2294, 1927, 1920, 23603, 2359, 11035, 6936, 1977, 2502, 16, 2679, 1920, 2428, 1927, 1920, 24192, 1930, 13647, 7655, 4966, 4712, 2032, 3261, 1922, 1920, 10688, 1009, 6235, 16650, 18, 8990, 16, 2338, 6042, 1927, 1920, 8091, 3374, 1922, 6751, 8548, 2094, 4041, 1920, 2529, 1927, 6665, 10258, 1930, 6386, 1920, 2529, 1927, 26424, 17182, 11677, 1019, 2094, 18, 11869, 1977, 2222, 3028, 2007, 3108, 10688, 1009, 5541, 2014, 8091, 3374, 2338, 6042, 18, 3, 2154, 3213, 2991, 3562, 8273, 11, 61, 10688, 2628, 35, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

b) context example after encoding

Figure 4.4 Encoding of context

4. Start/end token positions: The character start/end positions in the answer must be converted to token start/end positions. The built-in method char to token () is used to perform the conversion. *Figure 4.5* depicts the differences between start/end character positions and start/end token positions.



```
[('expression', (0, 10)),
 ('of', (11, 13)),
 ('the', (14, 17)),
 ('proapoptotic', (18, 30)),
 ('gene', (31, 35)),
 ('inhbb', (36, 41)),
 ('is', (42, 44)),
 ('increased', (45, 54)),
 (',', (54, 55)),
 ('while', (56, 61)),
 ('the', (62, 65)),
 ('levels', (66, 72)),
 ('of', (73, 75)),
 ('the', (76, 79)),
 ('antiapoptotic', (80, 93)),
 ('and', (94, 97)),
 ('oocyte', (98, 104)),
 ('maturation', (105, 115)),
 ('marker', (116, 122)),
 ('kit', (123, 126)),
 ('are', (127, 130)),
 ('decreased', (131, 140)),
 ('in', (141, 143)),
 ('the', (144, 147)),
 ('hes1', (148, 152)),
```

Figure 4.5: Left figure displays the index position of each word in the context, Right figure shows the token position of each word in the context.

5. Dataset for training: The Pytorch framework is used to prepare the dataset for model training. To prepare the data for training, we define a custom Dataset, which is used as input for training and evaluation in the following steps.

6.Training: Training begins once the model and dataset are complete. At this stage of the training, various parameters are defined.

Optimizer: AdamW

Learning rate: 1e-5

Batch_size: 16

Epoch:2

Device: GPU(‘cuda’)

7. Evaluation: The model is evaluated on the test set after training. Refer to *Figure 4.6* for an example of how the model generates start and end logits scores for each token present in each question context pair. These logits are used to predict the answer's start/end token position in the context. The expected start and end token positions are represented by the start and end token logits scoring the highest.

4.1.1 Evaluation metrics

1) *Accuracy (Exact Match):* It calculates the exact match of the actual start/end token to the predicted start/end token. EM is a binary measurement of whether the percentage of output from a system exactly matches the ground truth answer (the proportion of questions that are answered in exact same words as the ground truth) If Exactly matched Accuracy=1, otherwise, Accuracy=0.

2) *F1 score:* F1 is calculated using common words between the prediction and the true answer. The F1 score is based on precision and recall value. Precision is the proportion of shared words to the total number of words in the ground truth, whereas recall is the proportion of shared words to the total number of words in the prediction.

$$\boxed{F1=2^* [(precision^*recall) / (precision+recall)]}$$

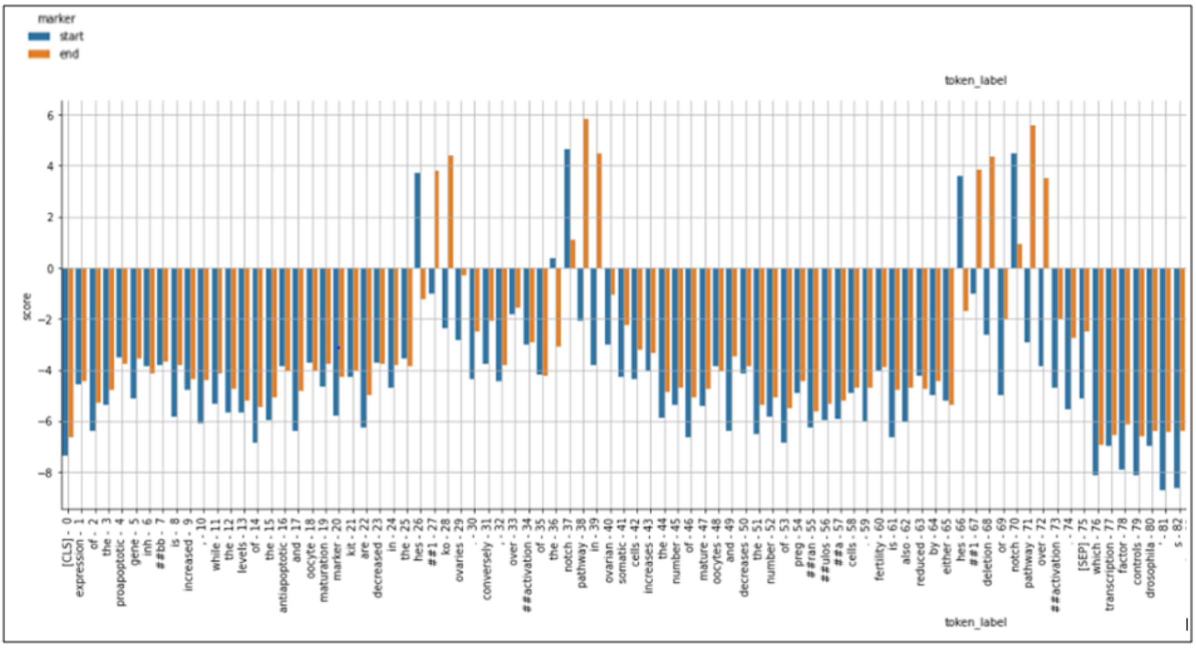


Figure 4.6: Displays values of start logits (blue bar) and end logits (orange bar) for each word in the context. X-axis is the token labels (tokens of context) and y-axis are the score of the logits

4.2 Proposed Span selection technique (start/end tokens)

When the testing dataset is evaluated on the trained model, the output generates the start and end logits score, which serves as a reference to calculate the predicted start/end token position, as mentioned in the previous section during Evaluation. The logit score for the start/end token position is clearly distinguished in *Figure 4.6*.

This start/end logit score predicts the answer text's start/end token position in the context provided in a sample. The prediction is chosen by taking the highest start/end logits score. Let us assume

‘HS(i)’ be the start logits score

‘HE(j)’ be the End Logits score

‘i’ be the token position for **start logits** representing each token label in the context such that 0<=i<n

(n –is the length of input tokens)

‘j’ be the token position for **end logits** representing each token label in the context such that 0<=j<n

(n –is the length of input tokens)

Suppose (i,j) is the predicted start/end token position which is determined from the highest logits score for start/end position respectively.

Predicted Start Position = Max [HS(i)] where (0<=i<n)

Max [HS (0), HS (1), HS (2), HS (3), ... HS(n-1)]

Predicted End Position = Max [HE(j)] where (0<=j<n)

Max [HE (0), HE (1), HE (2), HE (3), ... HE(n-1)]

Problem: The token sequence is ignored in the above method; for example, the start token position should always be less than the end token position to get the answer. The above statement can be modified as the following:

1.We first check the higher logits score between predicted start logits and predicted end logits

check max (HS(i), HE(j))

2. If predicted start logits is greater (HS(i) > HE(j)), then

Start token position (i) remains same

End token position (j) changes with respect to the start token position (i)

Max end logits score HE(j) is computed with revised token position which is greater than start token position(i) I.e., j ranges from (i+1<=j<n) Thus, we have

‘j’ end token position where {max [HE(j)] (i+1<=j<n)} condition is satisfied

3. If predicted end logits is greater (HS(i) < HE(j)), then

End token position (j) remains the same

Start token position (i) changes with respect to the start token position (i)

> Max start logits score HS(i) is computed with revised token position which is smaller than end token position (j) I.e., i range from (0<i<j) Thus, we have
>
> ‘i’ start token position where {max [HS(i)] (0<i<j)} condition is satisfied

Thus, (i, j) the predicted start/end token position satisfies (i<j) condition.

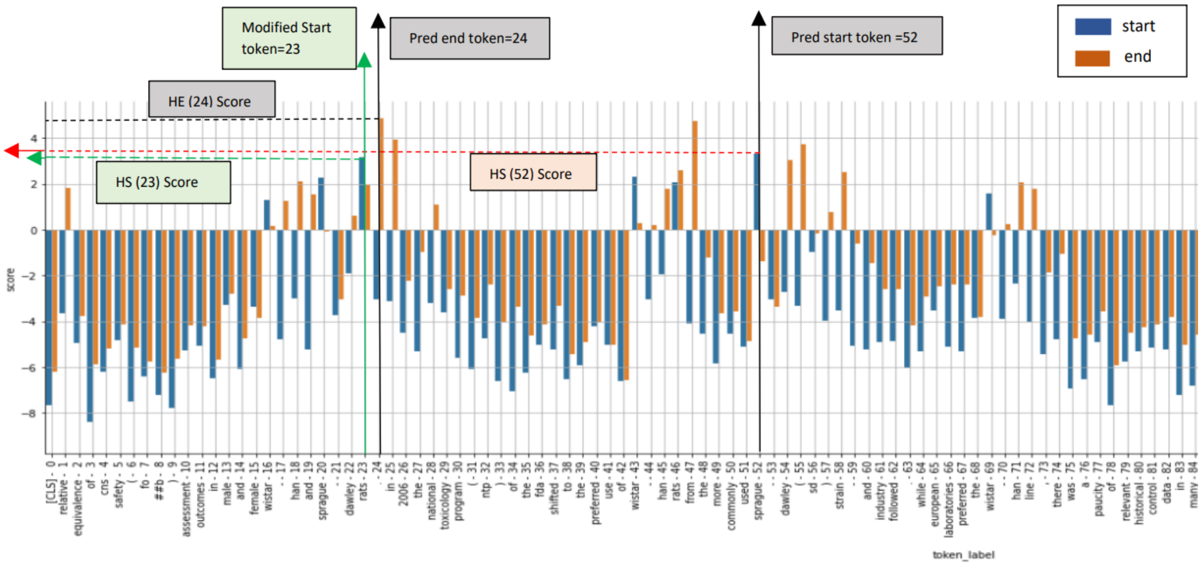


Figure 4.7: Example of start position > end position

Consider an example where initial results obtained are given below. Refer to *Figure 4.7* for details

Token	start	end
True	23	24
Pred	52	24

Here (i=52, j=24) is the predicted token positions which does not satisfy the condition (i<j)

We compare (HS (52), HE (24)) I.e., the predicted start/end token logits

End logits (HE (24)) is greater. Thus, end position j=24 remains unchanged

Start position (i) should be now based on max (HS(i)) where (0<i<24)

HS (23) is the highest score. Therefore, modified start position is i=23

Results after applying start/end sequence ordering:

Token start end

True 23 24

Pred 23 24

4.3 Dataset Augmentation

We have discussed the major points of the present work on question answering task by finetuning PubMed Bert trained on biomedical data. *Figure 4.2* shows an example of the question-answering dataset. The model is fed with question context pair where the input should account for a maximum of 512 tokens per example. The context is usually the abstract of the related document for each question present in the BioASQ dataset. For instance, on average, each question could be answered by eight relevant documents where a particular sentence or the entire abstract of a specific relevant document could be beneficial in predicting the correct answer.

Let us consider an example,

Question(q)

Relevant documents (r1, r2, r3, r4, r5, r6, r7, r8)

Each document abstract consists of five sentences (s1, s2, s3, s4, s5)

So, question (q) could be answered by context of r3(s3) or r3(s1, s2, s3, s4, s5) or other available options.

Instead of eight question context pairs, we could generate more question context pairs by varying the length of the abstract to the context. As a result, it is critical to comprehend the optimal context length for predicting correct answers to questions.

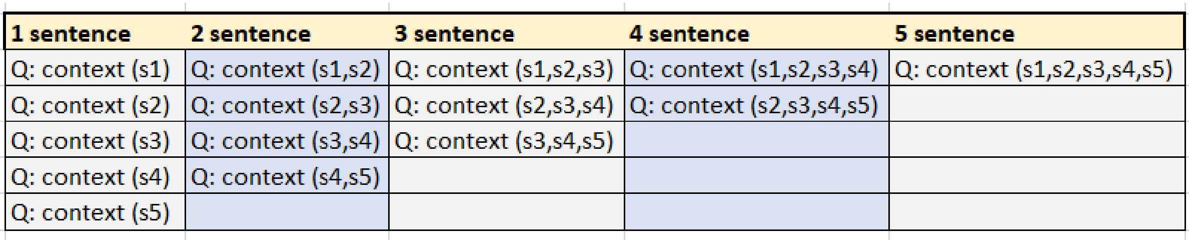
The context for the question is typically formed by the entire abstract of a relevant document. Data augmentation addresses the scarcity of examples for fine-tuning the model. Increasing the number of question context pairs by tailoring the context length would result in multiple contexts of an entirely abstract. Increasing the number of examples to train the model improves its accuracy in correctly predicting answers in the biomedical question-answering task.

4.3.1 Context of varying length

Creating a variable length context will help us understand how the model responds to changes in context length and an increase in the number of examples, which will help us in creating patterns of the changes observed in the accuracy to predict answers correctly.

As previously discussed, generating a variable context length from an abstract is critical to increasing the question context pair. Note we terminate the context by completing the sentence when creating variable context length. It will facilitate the preservation of the meaning of the words in the sentences. This is illustrated with the following example:

Question (Q) is associated with a relevant document who's abstract (a) consists of five sentences (s1, s2,s3,s4,s5). Establishing context with complete sentences that do not disrupt the sentence sequence is essential. As a result, we used the following technique.

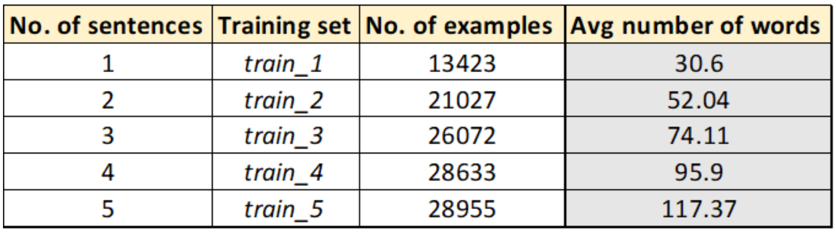


1 sentence	2 sentence	3 sentence	4 sentence	5 sentence
Q: context (s1)	Q: context (s1,s2)	Q: context (s1,s2,s3)	Q: context (s1,s2,s3,s4)	Q: context (s1,s2,s3,s4,s5)
Q: context (s2)	Q: context (s2,s3)	Q: context (s2,s3,s4)	Q: context (s2,s3,s4,s5)	
Q: context (s3)	Q: context (s3,s4)	Q: context (s3,s4,s5)		
Q: context (s4)	Q: context (s4,s5)			
Q: context (s5)				

Table 2: Examples with varying context length

4.3.2 Answerable Examples

Answerable question context pairs include the answer to the question within the context. Sometimes answer to the question may or may not be found in the passage. As a result, we build the examples using the technique illustrated in *Table 2* and select each context if it contains the answer to the question. Such instances are referred to as positive examples. In general, models are trained on positive examples to understand better the syntax, syntactic rules, and vocabulary involved in the biomedical domain, increasing the likelihood of domain understanding and accurate or near-perfect prediction of answers.



No. of sentences	Training set	No. of examples	Avg number of words
1	*train_1*	13423	30.6
2	*train_2*	21027	52.04
3	*train_3*	26072	74.11
4	*train_4*	28633	95.9
5	*train_5*	28955	117.37

Table 3: Answerable Training examples

No. of sentences	Testing set	No. of examples	Avg number of words
1	*test_1*	1956	31.23
2	*test_2*	3034	53.07
3	*test_3*	3716	75.64
4	*test_4*	4041	98.33
5	*test_5*	4055	120

Table 4: Answerable Testing examples

Based on the technique illustrated in *Table 2*, we create the training and testing sets (*Table 3* and *Table 4*) to increase the number of examples. It contains the context of varying lengths, represented by an average number of words. There were 1092 specific factual questions for training, with eight intermediate relevant documents for each question, for a total of approximately 8736 examples. The increase in the total number of samples generated by the varying length context method is depicted in *Table 4*.

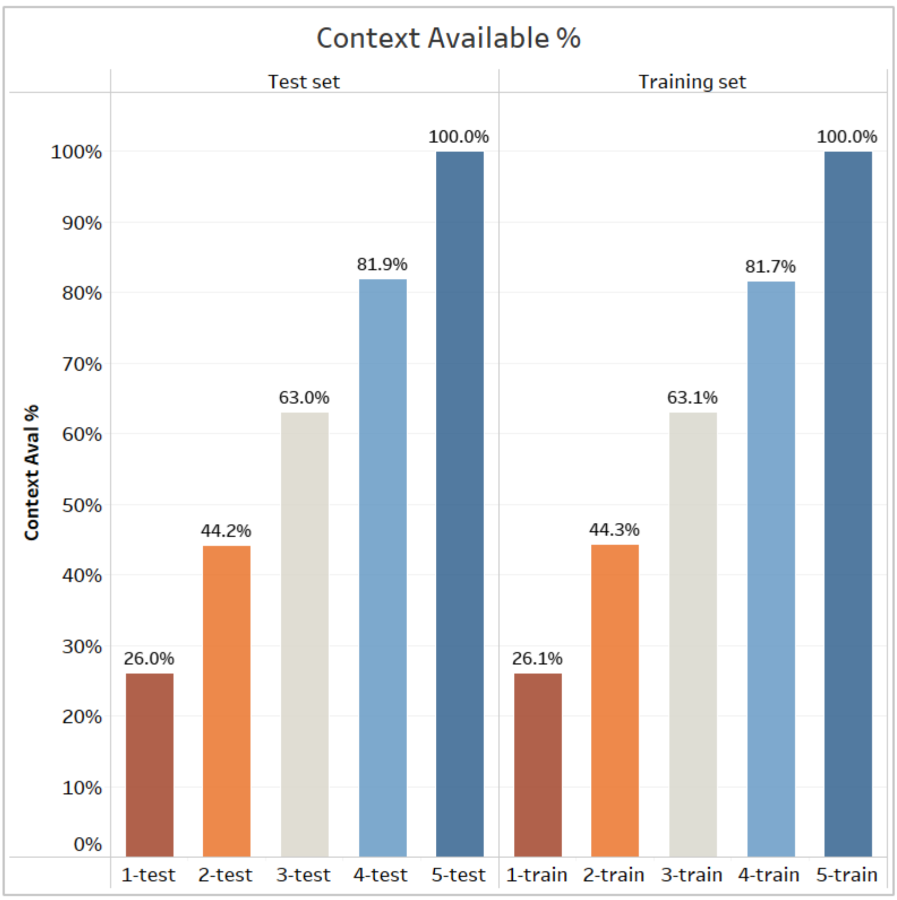


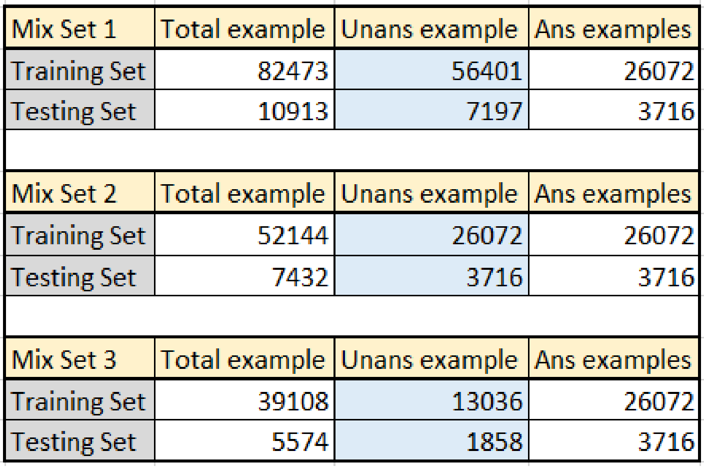
Figure 4.8: Training and Testing set context availability percentage

Furthermore, *Figure 4.8* illustrates the percentage of the context availability in terms of context length in the training and testing set. Creating a dataset with varying context availability percentages is critical for tracking model behaviour and establishing a pattern. It will help in improving the accuracy of predicting answers to the question by making the most out of the existing dataset.

4.4. Negative Samples

Question Answering Tasks typically focus on answerable/positive examples when training the model. The critical point is that given the question context pair, there is a reasonable chance that an answer not being present within the context. In this case, the model predicts a plausible explanation, which reduces its accuracy. The model should learn that not every question can be answered and anticipate unanswerable questions in such a situation.

Including negative samples in the dataset prepares the model for situations that are out of the ordinary and improves prediction accuracy. Traditionally, a model learns from an answerable example in which the model is aware that the answer is present in the context, which is advantageous for model prediction. Because the model is trained only on answerable standards, exposing it to various (positive/negative) testing examples reduces its accuracy in correctly predicting answers. *Table 5* gives total number of examples contained in mixed dataset.



Mix Set 1	Total example	Unans example	Ans examples
Training Set	82473	56401	26072
Testing Set	10913	7197	3716

Mix Set 2	Total example	Unans example	Ans examples
Training Set	52144	26072	26072
Testing Set	7432	3716	3716

Mix Set 3	Total example	Unans example	Ans examples
Training Set	39108	13036	26072
Testing Set	5574	1858	3716

Table 5: Total Number of examples in Mix dataset (Answerable/unanswerable)

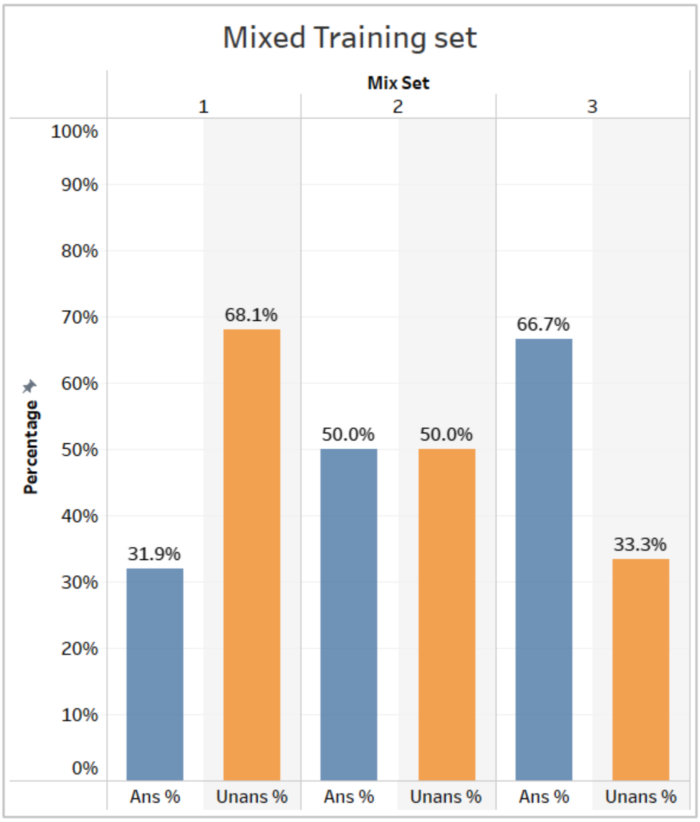


Figure 4.9: Training set consisting of Answerable and unanswerable examples

Thus, the addition of unanswerable examples strengthens the model's ability to handle any question and predict accurate or near-correct answers. Now that we know that adding negative samples is necessary, we need to determine an optimal number of negative examples for the desired accuracy output of the question-answering task. In order to do so, we generated three mixed datasets (positive and negative examples). The percentage of answerable and unanswerable samples in each dataset is shown in *Figure 4.9*.

Chapter 5

Experiments, Discussions, Comparison and Analysis

This chapter includes details about the experimental setup, including the technologies, tools, system configurations, model training and testing, PubMed Bert configurations and evaluation methods used to quantify the performance of the proposed approach. We will also present the results of the tests conducted using the dataset. To evaluate the effectiveness of the proposed approach, we have compared it with state-of-the-art methods. We present the performance by using the Bar and line graphs that will be showing the comparison based on the prediction Accuracy and F1 score obtained using each approach.

5.1 Tools and Libraries

We implemented our proposed approach using Pytorch framework with Python 3.7.15 programming language. The libraries and their versions that are used to implement our proposed approach are as listed below:

- Transformers
- torch 1.12.1+cu113
- Json 1.6.1
- NLTK 3.7
- Beautiful soup 4.6.3
- Pandas 1.3.5
- NumPy 1.21.6
- Seaborn 0.11.2
- Matplotlib 3.2.2
- requests 2.23.0

We used Jupyter notebook to implement and test the methods discussed in this work.

5.2 System Configurations

Data pre-processing, training of the model, and testing were done using the Jupyter notebook hosted on Google Colaboratory (also known as Google Colab). Google Colab is used to code and execute python scripts using browsers and can be used to implement machine learning and data analytics techniques. We used the Graphics Processing Unit (GPU) to empower the machine learning projects implemented using TensorFlow using custom-built processors by Google. Google Colab provides a GPU with 83.48 GB of RAM and 166.77 GB of memory.

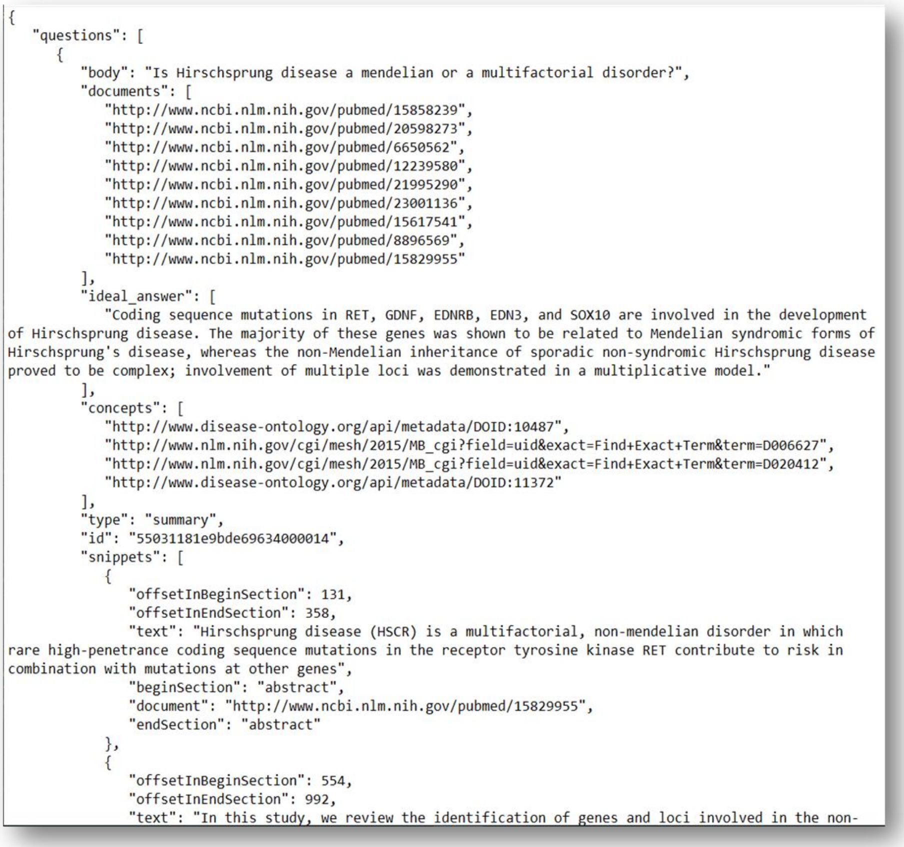
5.3 Dataset

The principal objective for our proposed method is to improve prediction accuracy in Biomedical Question Answering. Due to the availability of Squad Dataset, Question Answering was a huge success. To achieve comparable success in the biomedical domain dataset, a dataset that is as large as SQuad was required. Even though large biomedical datasets are available, unlike Squad, they are not manually generated. As a result, the generated answers are not guaranteed to be the golden answers. Therefore, in order to better suit our requirements for this task, the BioASQ dataset which happens to be the largest manually annotated biomedical dataset available was chosen. To bridge the gap, before applying the number of examples in the finetuning process, we first train the model with the Squad dataset, as Bert performs well with large datasets.

For the present thesis work, we have trained the PubMed Bert model on SQuad 2.0 and BioASQ 9b dataset.

SQuAD: The Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset comprised of questions posed by crowd workers on a set of Wikipedia articles, where the answer to each question is a segment of text, or span, from the corresponding reading passage, or the question may be unanswerable. SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowd workers to make them appear similar to answerable ones. To perform well on SQuAD, the system must answer questions when possible and must determine whether the paragraph supports no answer and it should refrain from providing answering.

BioASQ: BioASQ is an international biomedical challenge task comprising the information of the annual tasks on semantic indexing and biomedical question answering records [48]. Task 9b is a question-answering task designed for systems to answer four biomedical questions: factoid, summary, list, and yes/no. The participants are given questions as well as relevant snippets. Their systems generate either an exact answer (for yes/no, factoid, and list questions), an ideal answer (for summary questions), or both. The tasks are released in five batches over two months, with 24 hours to submit answers after each test batch. We focus primarily on factoid questions from the BioASQ9b dataset, which includes 1092 factoid questions out of a total of 3743 questions. *Figure 5.1* depicts an example of a question in the original dataset. Since the answer to the query is extracted from the relevant snippet, we consider the BioASQ challenge task to be an extractive QA task.



```
{
   "questions": [
      {
         "body": "Is Hirschsprung disease a mendelian or a multifactorial disorder?",
         "documents": [
            "http://www.ncbi.nlm.nih.gov/pubmed/15858239",
            "http://www.ncbi.nlm.nih.gov/pubmed/20598273",
            "http://www.ncbi.nlm.nih.gov/pubmed/6650562",
            "http://www.ncbi.nlm.nih.gov/pubmed/12239580",
            "http://www.ncbi.nlm.nih.gov/pubmed/21995290",
            "http://www.ncbi.nlm.nih.gov/pubmed/23001136",
            "http://www.ncbi.nlm.nih.gov/pubmed/15617541",
            "http://www.ncbi.nlm.nih.gov/pubmed/8896569",
            "http://www.ncbi.nlm.nih.gov/pubmed/15829955"
         ],
         "ideal_answer": [
            "Coding sequence mutations in RET, GDNF, EDNRB, EDN3, and SOX10 are involved in the development
of Hirschsprung disease. The majority of these genes was shown to be related to Mendelian syndromic forms of
Hirschsprung's disease, whereas the non-Mendelian inheritance of sporadic non-syndromic Hirschsprung disease
proved to be complex; involvement of multiple loci was demonstrated in a multiplicative model."
         ],
         "concepts": [
            "http://www.disease-ontology.org/api/metadata/DOID:10487",
            "http://www.nlm.nih.gov/cgi/mesh/2015/MB_cgi?field=uid&exact=Find+Exact+Term&term=D006627",
            "http://www.nlm.nih.gov/cgi/mesh/2015/MB_cgi?field=uid&exact=Find+Exact+Term&term=D020412",
            "http://www.disease-ontology.org/api/metadata/DOID:11372"
         ],
         "type": "summary",
         "id": "55031181e9bde69634000014",
         "snippets": [
            {
               "offsetInBeginSection": 131,
               "offsetInEndSection": 358,
               "text": "Hirschsprung disease (HSCR) is a multifactorial, non-mendelian disorder in which
rare high-penetrance coding sequence mutations in the receptor tyrosine kinase RET contribute to risk in
combination with mutations at other genes",
               "beginSection": "abstract",
               "document": "http://www.ncbi.nlm.nih.gov/pubmed/15829955",
               "endSection": "abstract"
            },
            {
               "offsetInBeginSection": 554,
               "offsetInEndSection": 992,
               "text": "In this study, we review the identification of genes and loci involved in the non-
```

Figure 5.1: Sample original factoid question

Pre-processing is used to convert the BioASQ dataset into the SQuAD format. A typical span-extractive question answering task gives the system a Context C and a question Q, and it must identify an answer span A (a_start, a_end) in C. The SQuAD dataset is an example of a span prediction QA task, which includes many question-answer pairs and a passage with answers to the given question. BioASQ's training dataset, on the other hand, contains a question, an answer, and multiple relevant snippets. As a result, we begin by pairing each snippet with its related question, resulting in many question-snippet pairs. In addition, based on the exact answer provided, we locate the answer's position in the snippet and populate it as the start position of the answer span in the snippet. During the pre-processing stage, we divided the snippets into several question-snippet pairs. *Figure 4.2* is a representation of pre-processed example of BioASQ.

5.4 PubMed BERT Configurations

```
Bert_For_Question_Answering
    vocab_size=30522,
    hidden_size=768,
    num_hidden_layers=12,
    num_attention_heads=12,
    intermediate_size=3072,
    hidden_act="gelu",
    hidden_dropout_prob=0.1,
    attention_probs_dropout_prob=0.1,
    max_position_embeddings=512,
    type_vocab_size=2,
    initializer_range=0.02,
    pad_token_id=0,
    gradient_checkpointing=False,
```

5.5 Comparison and Analysis

In this thesis work, we focused on answer prediction by (i) varying context length and (ii) Adding negative samples while using a new technique to determine the span of the answers in the context.

We propose the following techniques and compare the performance to the state of the art

- The method used for Finetuning SQuad 1.0 on BERT for answerable examples with varying context length
- The method used for Finetuning SQuad 2.0 on BERT for mixed sets, which include unanswerable examples
- Span Selection technique used in [49]

5.5.1 BERT On SQuad

In this thesis work, we used the method for finetuning BERT on the SQuad dataset [50] to finetune PubMed Bert on BioASQ as a baseline model to compare our results.

5.5.1.1 SQUAD 1.1 for answerable examples

We worked on the answerable examples of the BioASQ dataset in this downstream Question answering task. We represented the input question (consider as A) and paragraph (consider as B) pairs as a single packed sequence, using the A embedding for the Question and the B embedding for the paragraph. PubMed Bert model architecture *Figure 2.13* shows that a special token ([SEP]) separates the pairs. The model returns two logit tensors: one for the logits **(HS)** corresponding to the answer's start token and one for the logits ***(HE)*** corresponding to the answer's end token. The predicted answer span is calculated using start/end logits. In the original approach, a standard SoftMax procedure is used to compute the probability of word ***i*** being at the start and the end of the answer span. We used the start logits ***(HS)*** and end logits ***(HE)*** scores for prediction. Following that, we use the sum of the scores ***HS(i) + HE(j) as*** a candidate span from position ***i*** to position ***j*** and the maximum sum of the scoring span where **(j ≥ *i*)** is used as a prediction for span selection. For the span detection our baseline model, we use the technique described in [49].

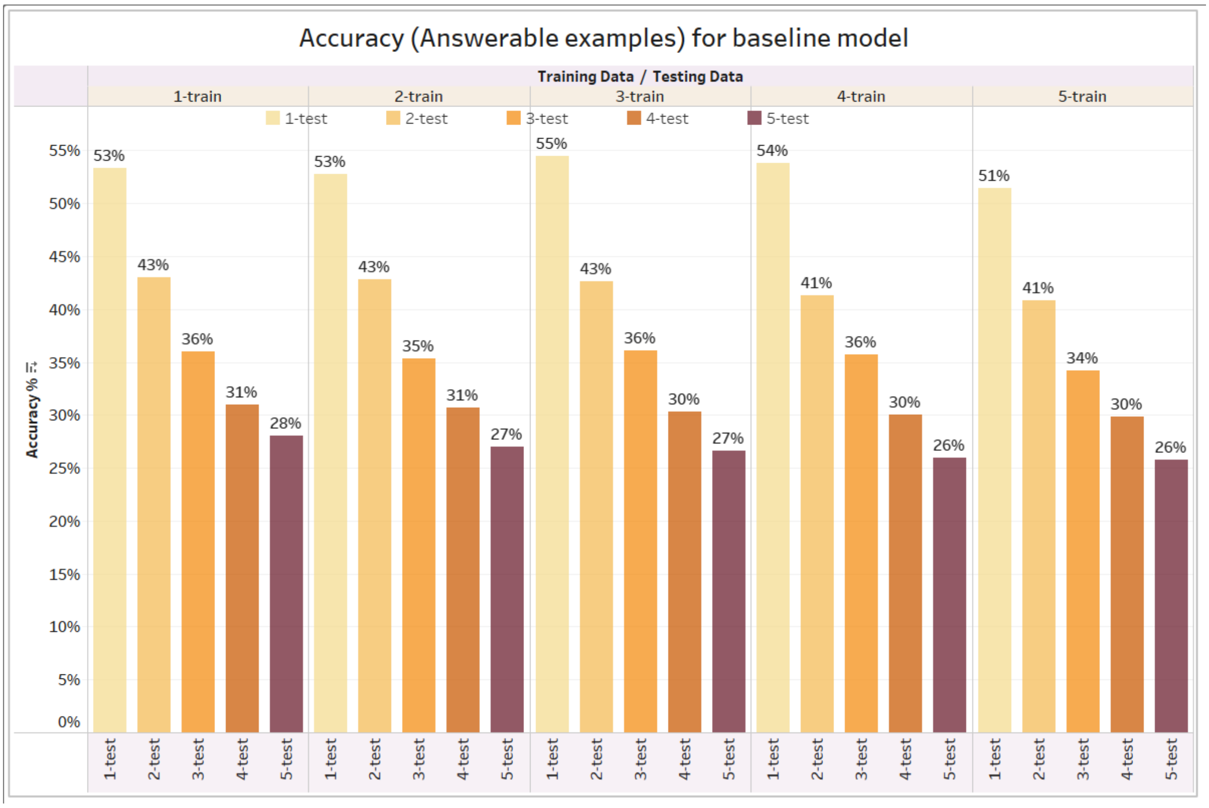


Figure 5.2: Accuracy of the baseline model with varying context length

To train the PubMed Bert, we first use the squad dataset and then the BioASQ dataset, as described above. Using the technique shown in *Table 2*, we created five different datasets with varying context lengths. After the training stage, we evaluated the model with various test sets. The accuracy obtained on these test sets is shown in *Figure 5.2*, and a bar graph demonstrates the wide range of accuracies obtained on different test sets with the same training set. It shows pattern that regardless of the length of the context in the training set, the test set performs similarly. The variation of the context length during the model training is less significant than that of the context length during the testing.

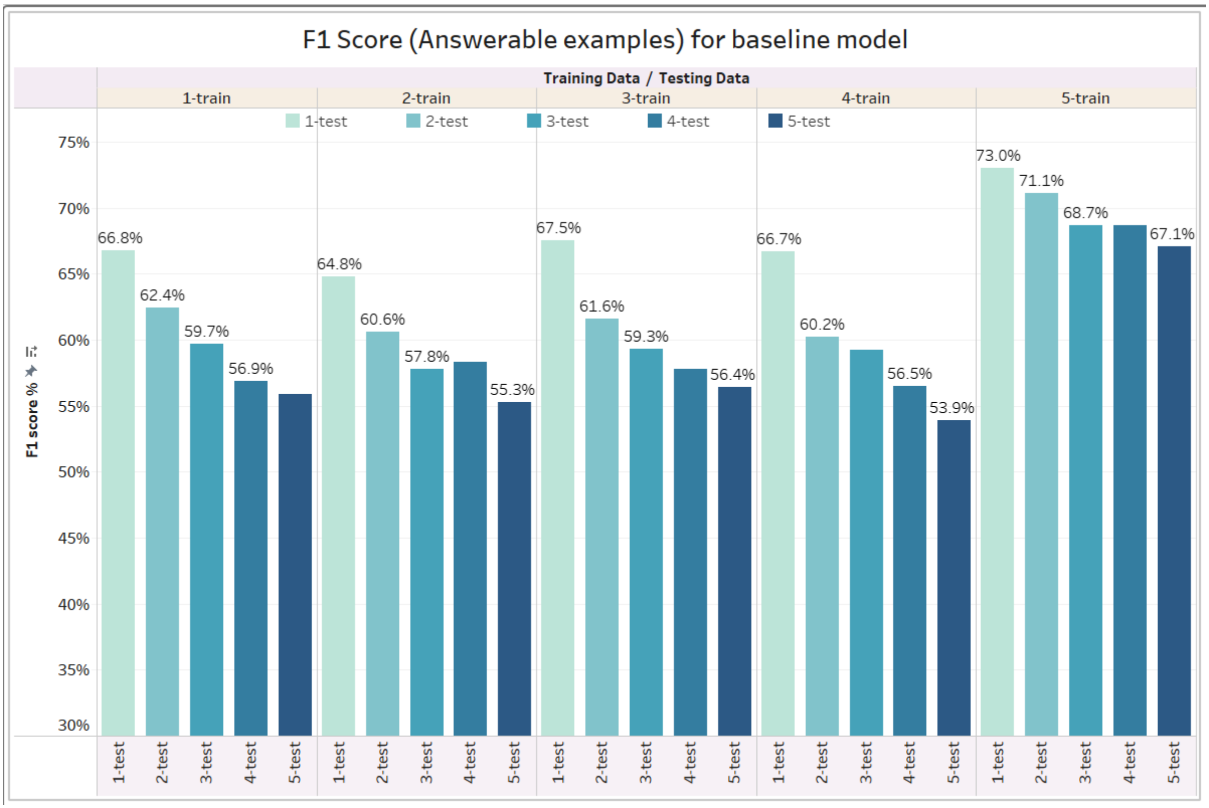


Fig 5.3: F1 score of the baseline model with varying context length

Figure 5.3 shows a similar trend for the F1 score of the prediction, with an increase in the F1 score from the 5-test set (with context of 5 sentences) to the 1-test set (with context of 1 sentence). The F1 score takes into account the tokens present in the answer prediction rather than just the index position of the start/end token. A significant difference is observed comparing the accuracy and the F1 score for the 5-test set. The availability of large number of tokens in a 5-test set (with extended context length) provides better possibilities for predicting an answer. In the 1-test, the availability of a small number of tokens reduces the likelihood of incorrect answer prediction. In contrast to accuracy, the F1 score captures most answer tokens I.e., tokens representing word in the correct answer even if the start/end position is mispredicted. For example, the index of the correct token (start=24, end=30) and predicted tokens index (start=25, end=31) are off by a few token positions, but they still contain common tokens in the answer span which are considered in the F1 score calculation.

Upon evaluation it was observed that, the smaller the context of the given question context pair, the higher the accuracy and F1 score to predict correct answers. It could be because the model deals with lesser token, and the probability of predicting incorrect answers decreases drastically than those with large context length. The accuracy and F1 score demonstrate the behaviour of the models on varying context length of the samples.

5.5.1.2 SQUAD 2.0 with mixed examples

We use the method described in [50] for Squad 2.0 training. We also treat questions that do not have an answer. This is handled by having an answer span with start and end index at the [CLS] tokens. The position of the [CLS] token is considered for the start and end answer span positions. For prediction, we compare the score of the no-answer span: ***snull = HS (0) + HE (0)*** to the score of the best non-null span ***S (i, j) =HS(i) + HE(j) (i,j >0)***. We predict a non-null answer when ***S (i, j) > snull***.

We use the same model training procedure for answerable examples (SQUAD 1.1) to get start/end logits and span selection for start and end index, with a few changes to accommodate unanswerable examples.

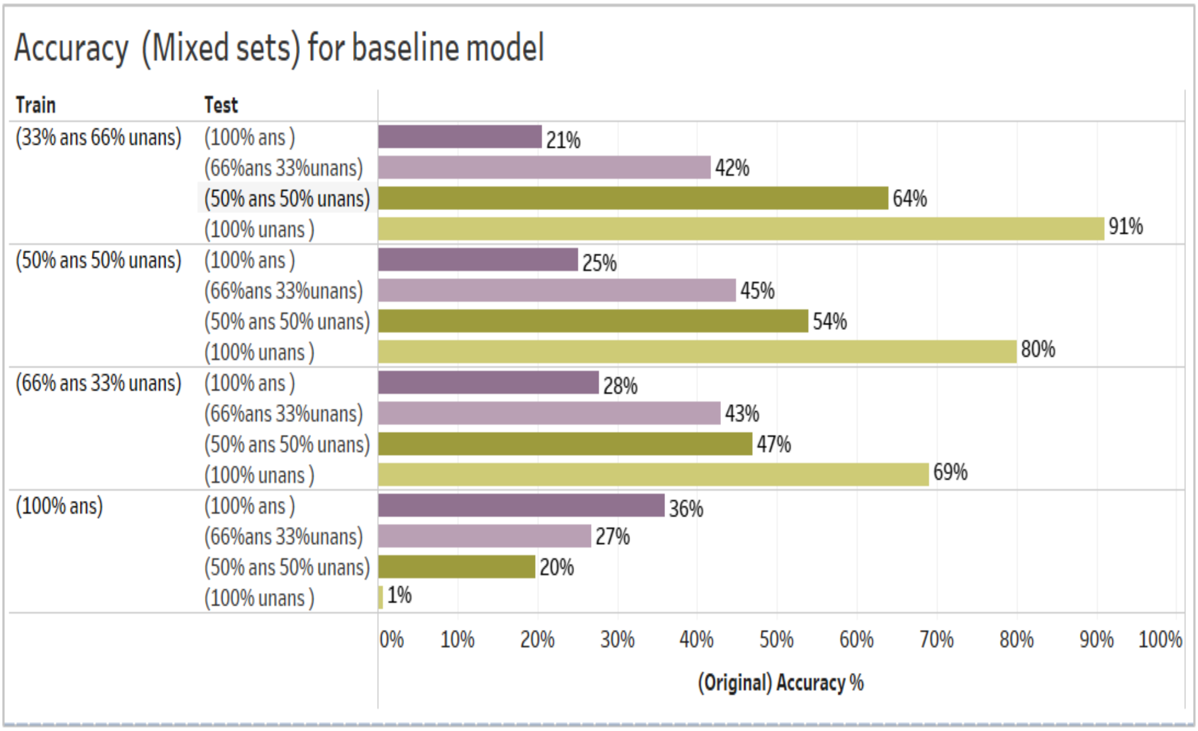


Figure 5.4: Accuracy with mixed examples

To understand the impact of unanswerable examples on model behaviour, we created mixed training and testing sets of answerable and unanswerable examples in various ratios. *Figure 5.4* depicts the accuracy of different mixed test sets on training sets, while *Figure 5.5* depicts the F1 scores.

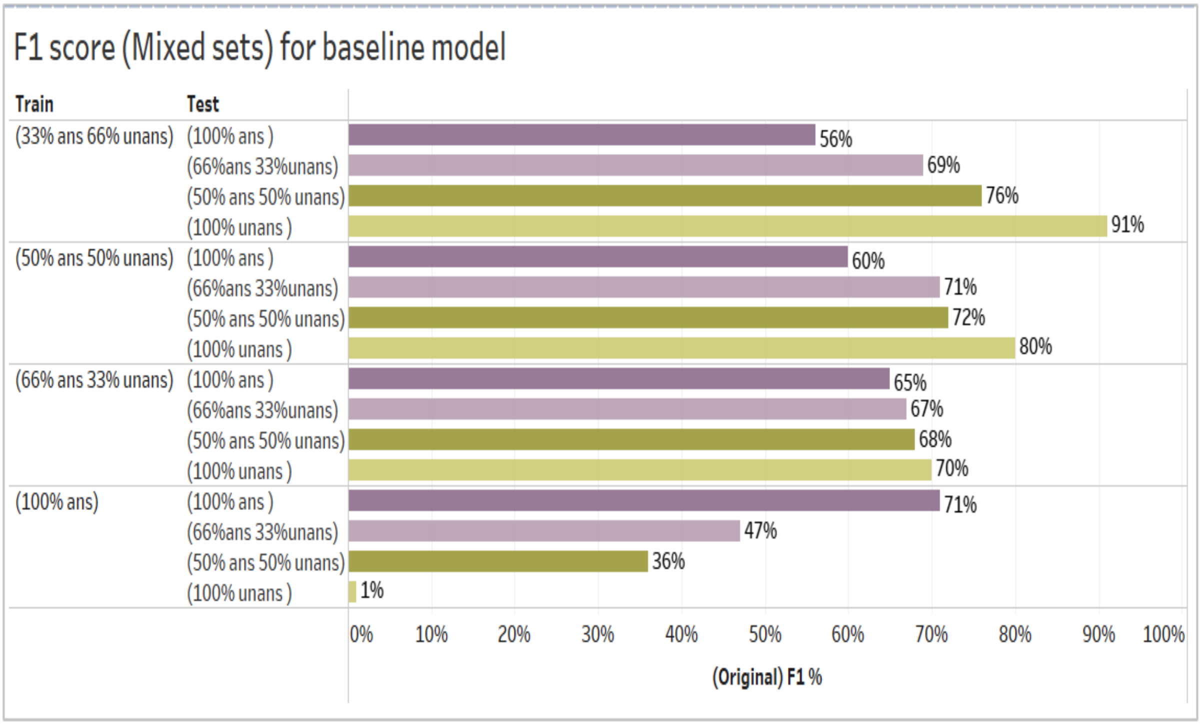


Figure 5.5: F1 score with mixed examples

The accuracy for 100% answerable test set is highest when trained on 100 % answerable training set. Accuracy decreases as the percentage of unanswerable examples in training set increase. As noticed in *figure 5.4* the decrease in accuracy is minimum. Thus, addition of negative samples in the training samples shows low impact on accuracy fall while increasing the robustness of the model.

100% unanswerable test set shows 1% percent of accuracy for model trained on 100% of answerable samples. This occurs because the model is unaware of no answer possibility and gives wrong predictions. Figure 5.4 shows as the percent of unanswerable examples in the training set rise there is a sharp increase in accuracy. We could say that the model learns well when trained on negative samples.

The accuracy obtained from first three test sets for model trained on each training set, on positive example prediction, reflect the trend observed for 100% answerable test set. Similarly, the accuracy obtained from last three test sets for model trained on each training set, on negative example prediction reflect the trend observed for 100% unanswerable test set.

Even though accuracy and F1 score display matching trend, their values show noticeable differences. The difference is much higher in test sets containing a more significant number of answerable examples compared to unanswerable ones. This is due to the fact that the F1 score considers the common word inside the golden answer and predicted answer, unlike accuracy, which strictly calculates based on the start and end index position.

When trained on mixed examples, the model struggles to predict the accuracy of answerable examples (100% answer). It happens because unanswered prediction is much easier to make. The model requires much more effort for the answered prediction, followed by the answer's location within the context span and then the index position of the start and end token. As a result, a suitable ratio of answerable and unanswerable examples for training is essential in fulfilling our goal.

5.5.2 Span Selection technique

We use a heuristic way [49] to select the answer span at the time of prediction. Firstly, we get 20 most probable start positions and 20 most probable end positions (i.e., Logits score of highest start and end position). Then for the $20 * 20$ start-end position pairs, we remove invalid logit pairs, (i.e., a pair with start and end positions): start position after end position; start to end span exceeds the max answer length we set. We choose the one with the highest ***pstart(i) + pend(j)*** as a candidate answer from the valid pairs. We use ***pstart (0) + pend (0)*** as the question's unanswerable score. If ***pstart (0) + pend (0) > pstart(i) + pend(j)***, then we predict this question as unanswerable; otherwise, we expect the candidate's answer.

We used the above-mentioned selection technique for the baseline model, and implemented our proposed technique explained in *Table 2* as an improvement and compared the results. The new technique is applied to answerable examples (with varying context lengths) and a set of mixed examples.

5.5.2.1 For answerable examples (variable context length)

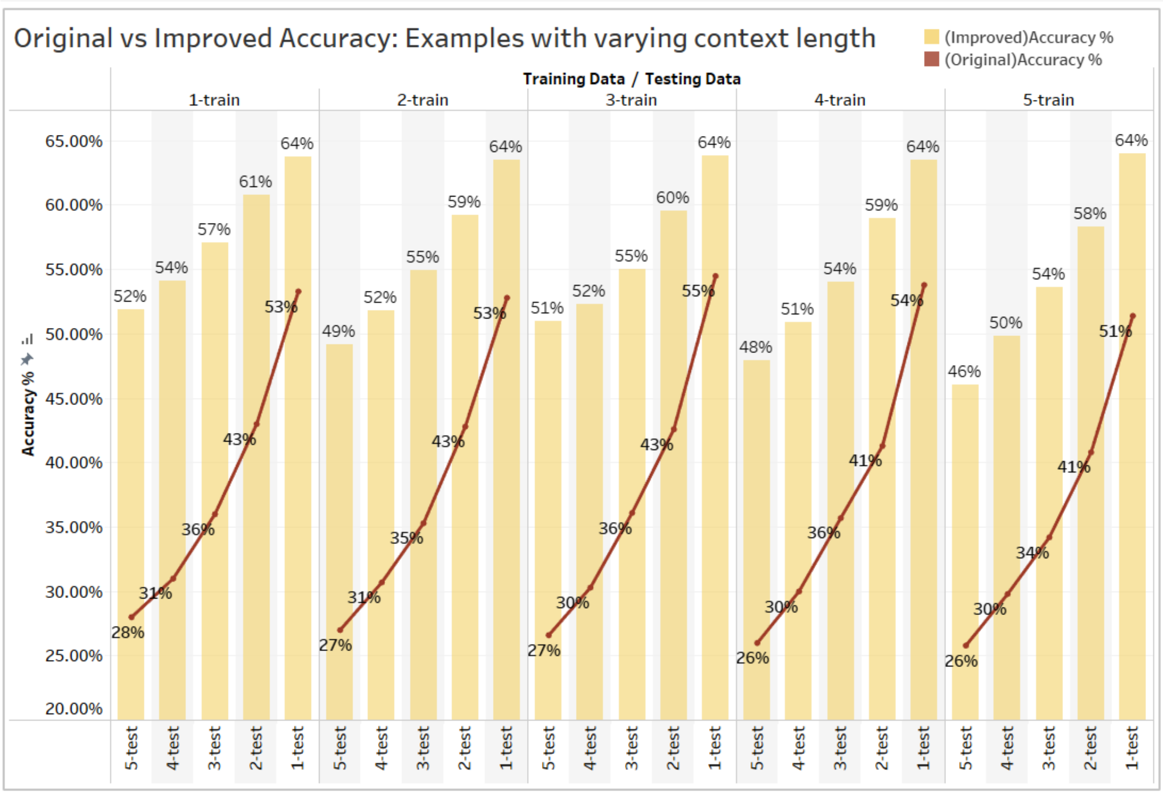


Figure 5.6: Comparison of Original and Improved Accuracy (Examples with Varying context length)

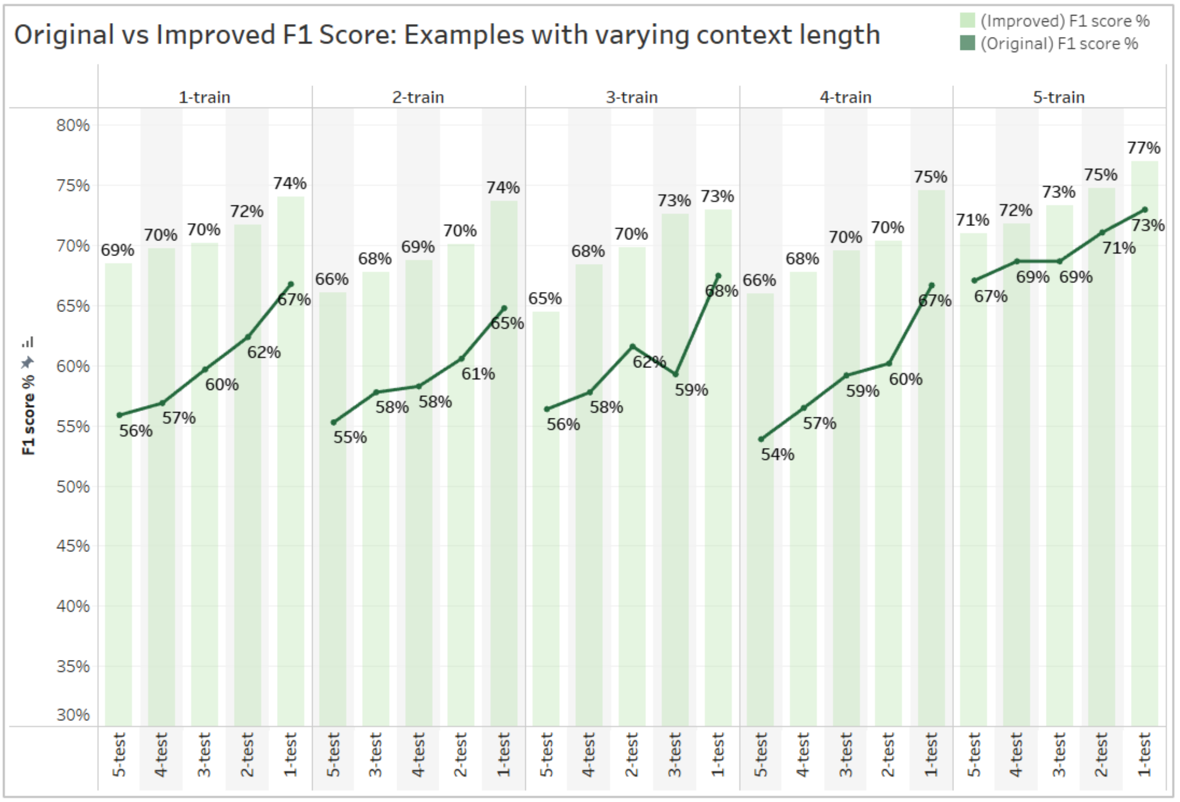


Figure 5.7: Comparison of Original and Improved F1 score (Examples with Varying context length)

As observed in *Figure 5.6*, the score for improved accuracy is higher than the score for original accuracy. A significant improvement can be seen for a 5-test set. As the length of the test set decreases, so does the improved accuracy with fewer tokens. Since 1-test set contains lesser tokens resulting in a lower chance of improvement of the previous technique. *Figure 5.7* depicts a similar pattern with the F1 score.

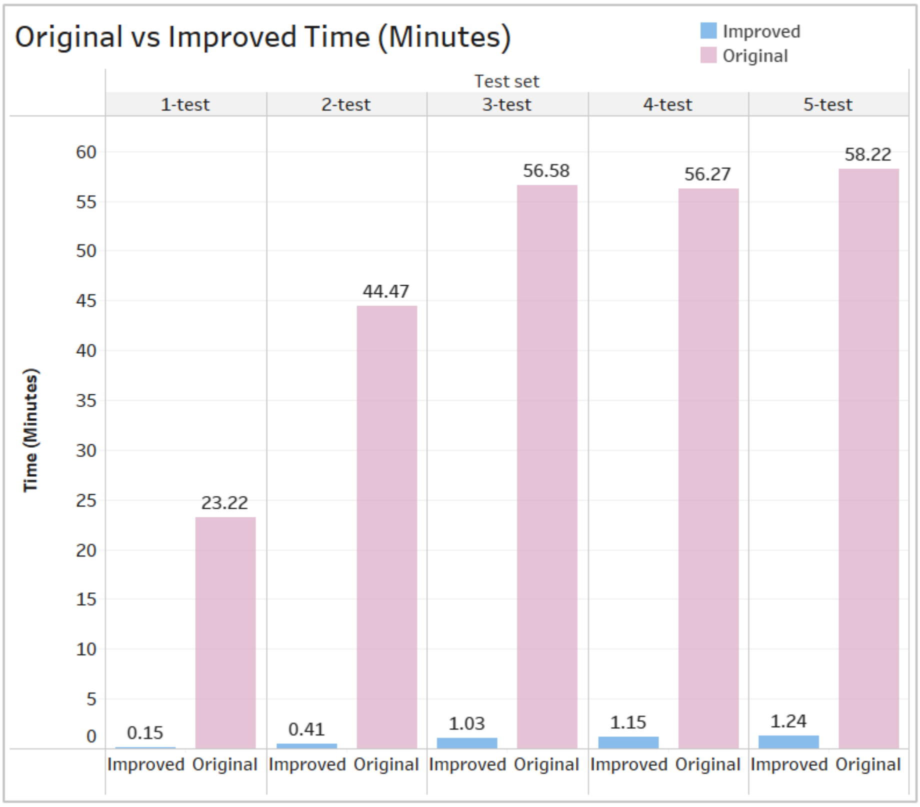


Figure 5.8: Comparison of Original and Improved Time (Examples with Varying context length)

Figure 5.8 compares the average time (in minutes) taken to evaluate the test sets when the model is trained on multiple training sets. We see an apparent rift in values because previous techniques capture the top 20 start and end logits. It also evaluates valid start and end pairs, calculates their paired scores and selects the topmost pair of scores. This process is performed for each example in the test set, consuming much of the processing time.

5.5.2.2 For mixed sets

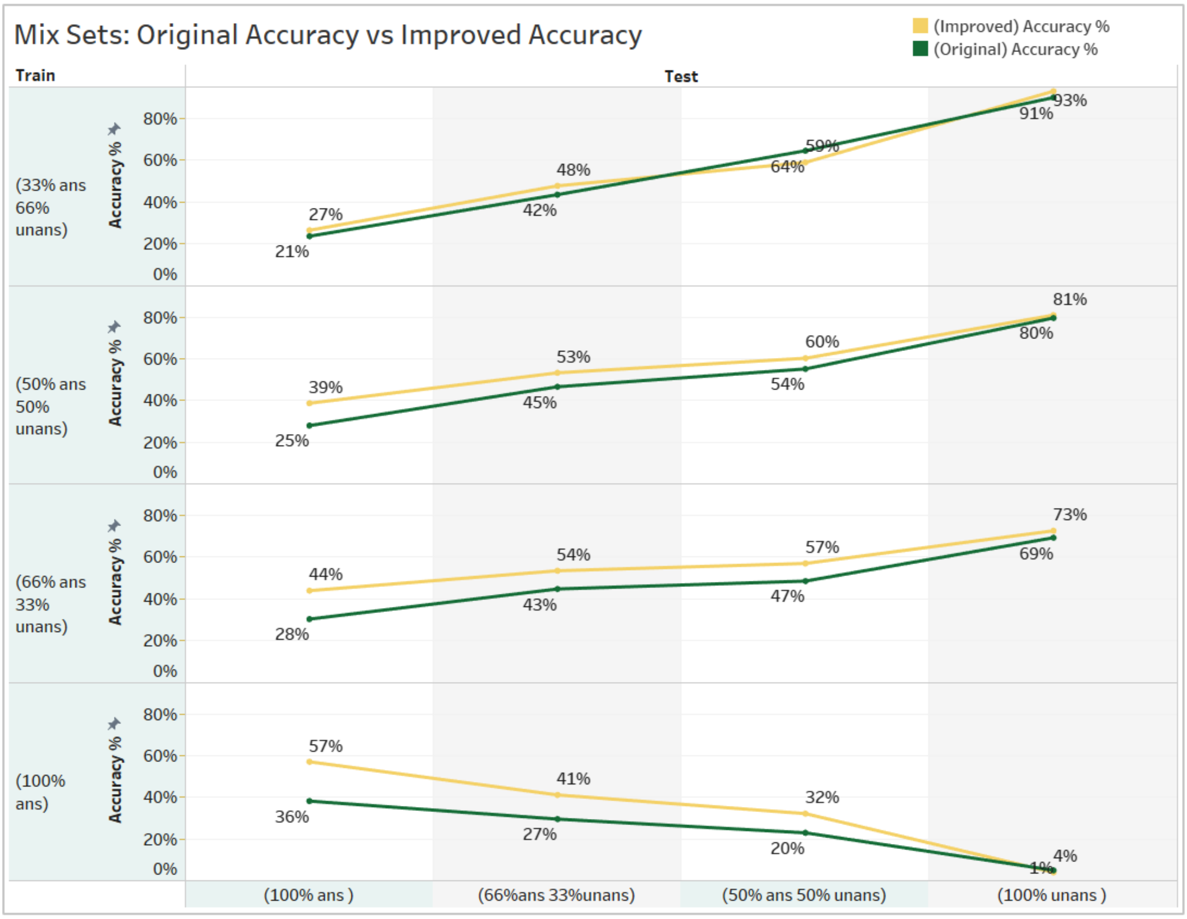


Figure 5.9: Comparison of Original and Improved Accuracy (Mixed sets)

In *Figure 5.9* we can see that there is an increase in accuracy when the percentage of answerable examples is greater than the percentage of unanswerable examples in training sets. Improvement in accuracy and F1 score (*Figure 5.10*) is observed for test sets containing a greater number of answerable examples. Contrarily, there is little or no room for improvement with model results, when tested on multiple unanswerable examples. This shows that the improvement for answerable prediction is higher than unanswerable ones.

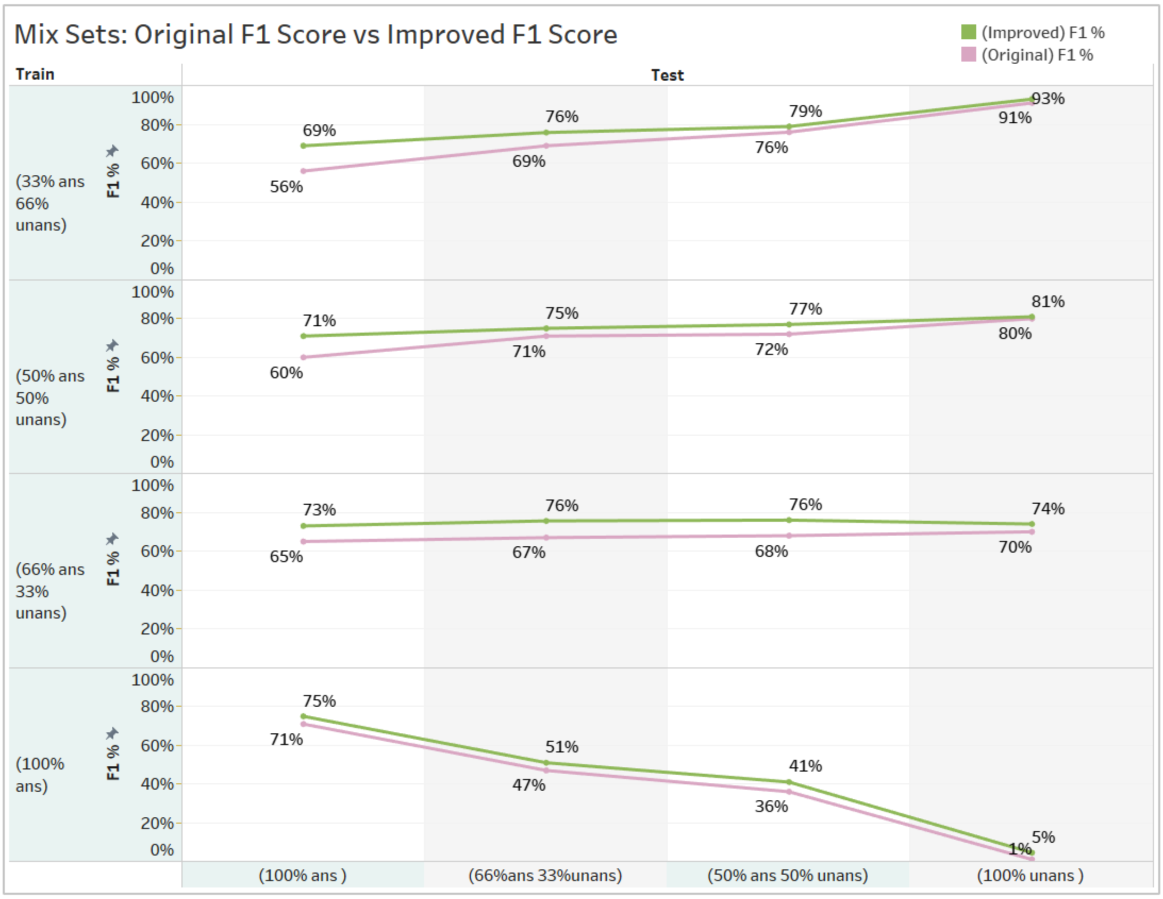
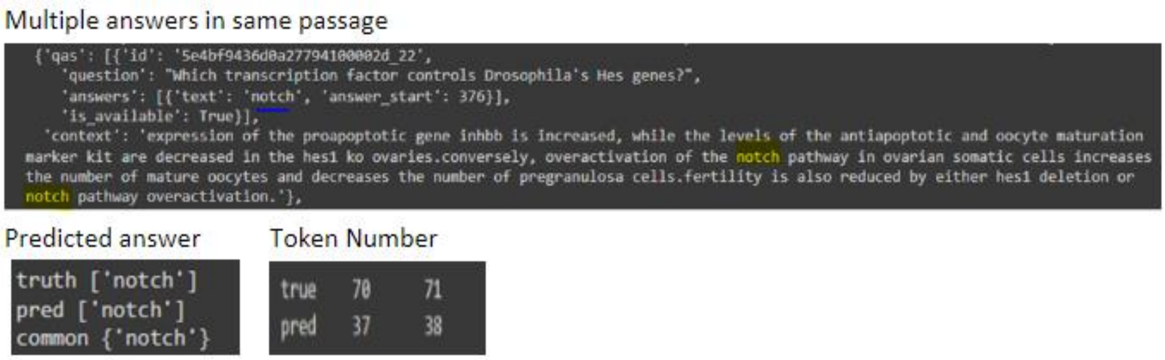


Figure 5.10: Comparison of Original and Improved F1 score (Mixed sets)

5.6.3 Post-Processing step

In the present work, while calculating the accuracy of the index position, we have considered the possibility of multiple answers available in the same context (with considerable context length, i.e., 5-test). For accuracy calculation, we do token matching of the predicted and actual answers and compare the index position. It helps to improve the accurate prediction of an answer.



Multiple answers in same passage

```
 {'qas': [{'id': '5e4bf9436d0a27794100002d_22',
    'question': "Which transcription factor controls Drosophila's Hes genes?",
    'answers': [{'text': 'notch', 'answer_start': 376}],
    'is_available': True}],
   'context': 'expression of the proapoptotic gene inhbb is increased, while the levels of the antiapoptotic and oocyte maturation
marker kit are decreased in the hes1 ko ovaries.conversely, overactivation of the notch pathway in ovarian somatic cells increases
the number of mature oocytes and decreases the number of pregranulosa cells.fertility is also reduced by either hes1 deletion or
notch pathway overactivation.'},
```

Predicted answer

```
truth ['notch']
pred ['notch']
common {'notch'}
```

Token Number

```
true   70    71
pred   37    38
```

5.11: Example of Multiple answers scenario

Chapter 6

Conclusion and Future Work

6.1 Conclusion

With the world on lockdown due to pandemics such as Covid-19, accurate biomedical information about disease symptoms, precautions, and prevention is critical. Platforms such as Biomedical question answering play an important role in disseminating this information to people living in remote areas. To meet the demand, we need a dependable question-answering system that can understand the context of each question and respond as accurately as possible. Thus, a large number of lives are saved, or at the very least, the professional's time.

A biomedical question-and-answer system is being researched, and efforts to improve it are ongoing. However, we are still a long way from having a reliable system. Deep learning model like BERT shows promising results on Natural Language processing task like Question answering. However, there are still obstacles to overcome, such as the lack of large expert-annotated biomedical datasets. Model performance on negative samples, as well as the optimal performance of the original question-answering technique.

This thesis work proposes a solution to the difficulties associated with answering biomedical questions. To begin, the overall dataset size is increased by augmenting the available largest expert annotated biomedical dataset. In addition, we investigated the model's performance on examples with varying context lengths. It aided us in understanding how the model behaves when the context length shrinks or expands. Even though most of the answers are available in the corpora, there are still questions that cannot be answered; therefore, instead of providing plausible answers, the model should be aware of the unanswerable situation. To deal with such a scenario, it was necessary to train the model with mixed samples and capture the model performance for different ratios of positive and negative samples in the training set.

We propose a new technique for predicted answer span selection, i.e., start and end token position. We applied both original and proposed techniques to the model being trained using two different dataset approaches (answerable dataset with varying context length and Mixed example dataset). The results show that the accuracy and F1 score percentages have improved.

6.2 Discussion

This study looked at the model's performance on different datasets, such as those with varying context lengths and mixed examples. The goal of using datasets with varying context lengths was to increase the small size of the dataset by augmenting the existing one. To accomplish this, we needed a strategy that would preserve the model architecture while outperforming existing results. Varying context length was an attempt to investigate how dynamic changes in passage length affect model performance. This provided us with a better understanding of the optimal length that performs well and the length where the model confuses and predicts incorrectly.

With the results obtained, we moved one step closer to understanding how the model responds to negative samples. To test the model's understanding of language, particularly the question-answering task, if an unanswerable question is encountered. The experiments demonstrated that it predicts plausible answers because the model must learn that a non-answer is an option when answering questions. To learn about the behaviour of the models, we created Mixed sets with different answers and unanswerable ratios. Experiments demonstrate a range of behaviour on different training and testing sets. We could say that a threshold for ratio calculation is needed for training sets which can be defined based on one's prediction requirement.

While studying the model performance on the baseline model, we discovered that model evaluation took a long time. Further investigation revealed that the span selection technique was being used while answer prediction was taking up a significant amount of time. we proposed a technique to improve the answer prediction rate thereby reducing the evaluation time. The model trained using the proposed span selection technique, shows promising results.

6.3 Future Work

We observed various model behaviour of datasets with varying context lengths and mixed examples in our experiments. The pattern discovered can be applied to larger problems in question answering. It is possible to incorporate a better technique for unanswerable prediction. Knowing the question-and-answer type requirements while developing different techniques can be a game changer. While evaluating, this thesis proposed a span selection technique that can be expanded to other domains. This thesis concentrated on factual biomedical questions; however, similar research with other question types, particularly list questions, is important. For promising results, generative question answering using summary questions should be researched and explored. Furthermore, when considering the biomedical domain, the emphasis on knowing the biomedical term, its usage, and prediction pattern must be studied to improve prediction results.

BIBLIOGRAPHY

[1] Jin, Q., Yuan, Z., Xiong, G., Yu, Q., Ying, H., Tan, C., ... & Yu, S. (2022). Biomedical Question Answering: A Survey of Approaches and Challenges. ACM Computing Surveys (CSUR), 55(2), 1-36.

[2] Longpre, S., Lu, Y., Tu, Z., & DuBois, C. (2019). An exploration of data augmentation and sampling techniques for domain-agnostic question answering. arXiv preprint arXiv:1912.02145.

[3] 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.

[4] Jeong, M., Sung, M., Kim, G., Kim, D., Yoon, W., Yoo, J., & Kang, J. (2020). Transferability of natural language inference to biomedical question answering. arXiv preprint arXiv:2007.00217.

[5] Transformer: A Novel Neural Network Architecture for Language Understanding: https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

[6] Ishwari, K. S. D., Aneeze, A. K. R. R., Sudheesan, S., Karunaratne, H. J. D. A., Nugaliyadde, A., & Mallawarrachchi, Y. (2019). Advances in natural language question answering: A review. arXiv preprint arXiv:1904.05276.

[7] Ojokoh, B., & Adebisi, E. (2018). A review of question answering systems. Journal of Web Engineering, 17(8), 717-758.

[8] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240.

[9] 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.

[10] Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323.

[11] Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pre-trained language model for scientific text. arXiv preprint arXiv:1903.10676.

[12] Peng, Y., Yan, S., & Lu, Z. (2019). Transfer learning in biomedical natural language processing: an evaluation of BERT and Elmo on ten benchmarking datasets. arXiv preprint arXiv:1906.05474.

[13] Michalopoulos, G., Wang, Y., Kaka, H., Chen, H., & Wong, A. (2020). Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus. arXiv preprint arXiv:2010.10391.

[14] He, Y., Zhu, Z., Zhang, Y., Chen, Q., & Caverlee, J. (2020). Infusing disease knowledge into BERT for health question answering, medical inference and disease name recognition. arXiv preprint arXiv:2010.03746.

[15] Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., ... & Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH), 3(1), 1-23.

[16] Soni, S., & Roberts, K. (2020, May). Evaluation of dataset selection for pre-training and fine-tuning transformer language models for clinical question answering. In Proceedings of The 12th Language Resources and Evaluation Conference (pp. 5532-5538).

[17] Ouyang, K. SQuAD to BioASQ: analysis of general to specific.

[18] Pampari, A., Raghavan, P., Liang, J., & Peng, J. (2018). Emrqa : A large corpus for question answering on electronic medical records. arXiv preprint arXiv:1809.00732.

[19] Suster, S. and Daelemans, W. (2018). CliCR : A Dataset of Clinical Case Reports for Machine Reading Comprehension. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, pages 1551–1563. Association for Computational Linguistics.

[20] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392. Association for Computational Linguistics.

[21] L. Kodra and E. Kajo, "Question Answering Systems: A Review on Present Developments, Challenges and Trends", International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, 2017 [Online]. Available: https://thesai.org/Downloads/Volume8No9/Paper_31-Question_Answering_Systems_A_Review_on_Present_Development s.pdf. [Accessed: 22- May-2018].

[22] H. Madabushi and M. Lee, "High Accuracy Rule-based Question Classification using Question Syntax and Semantics", Aclweb.org, 2018. [Online]. Available: http://www.aclweb.org/anthology/C16- 1116. [Accessed: 23- May- 2018].

[23] E. Riloff and M. Thelen, "A Rule-based Question Answering System for Reading Comprehension Tests", 2018. [Online]. Available: https://pdfs.semanticscholar.org/4454/06b0d88ae965fa587cf5c167374 ff1bbc09a.pdf. [Accessed: 23- May- 2018].

[24] S. K. Dwivedia and V. Singh, "Research and reviews in question answering system," in Proceedings of International Conference on Computational Intelligence: Modeling Techniques and Applications, 2013, pp. 417 – 424.

[25] Muthutantrige, S. R., and Weerasinghe, A. "Sentiment Analysis in Twitter messages using constrained and unconstrained data categories," in Proceedings of the Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2016. doi:10.1109/icter.2016.7829935

[26] T. Mikolov and G. Zweig, "Context Dependent Recurrent Neural Network Language Model", 2012. [Online]. Available: https://www.microsoft.com/en-us/research/wpcontent/uploads/2012/07/rnn_ctxt_TR.sav_.pdf. [Accessed: 24- May2018].

[27] T. Mikolov, S. Kombrink, L. Burget, J. Cernock ˇ y and S. Khudanpur, "Extensions of recurrent neural network language model," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011.

[28] M. Tan, C. Santos, B. Xiang and B. Zhou, "LSTM-based Deep Learning Models for Non-factoid Answer Selection", 2015. [Online]. Available: https://arxiv.org/abs/1511.04108. [Accessed: 24- May2018]

[29] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies; 2018. p. 2227–2237

[30] Balikas, G.; Krithara, A.; Partalas, I.; Paliouras, G. BioASQ: A challenge on large-scale biomedical semantic indexing and question answering. In International Workshop on Multimodal Retrieval in the Medical Domain; Springer: Berlin/Heidelberg, Germany, 2015; pp. 26–39. [CrossRef]

[31] Xiong, C.; Su, M. IARNN-Based Semantic-Containing Double-Level Embedding Bi-LSTM for Question-and-Answer Matching. Comput. Intell. Neurosci. 2019, 2019, 6074840. [CrossRef]

[32] Dina, D.F.; Yassine, M.; Asma, B.A. Consumer health information and question answering: Helping consumers find answers to their health-related information needs. J. Am. Med. Inform. Assoc. 2020, 27, 194–201. [CrossRef]

[33] Kolomiyets, O.; Moens, M.F. A survey on question answering technology from an information retrieval perspective. Inf. Sci. 2011, 181, 5412–5434. [CrossRef]

[34] Heie, M.H.; Whittaker, E.W.; Furui, S. Question answering using statistical language modelling. Comput. Speech Lang. 2012, 26, 193–209. [CrossRef]

[35] Seena, I.; Sini, G.; Binu, R. Malayalam question answering system. Procedia Technol. 2016, 24, 1388–1392. [CrossRef]

[36] Goodwin, T.R.; Harabagiu, S.M. Knowledge representations and inference techniques for medical question answering. ACM Trans. Intell. Syst. Technol. 2017, 9, 1–26. [CrossRef]

[37] Kodra, L.; Meçe, E.K. Question answering systems: A review on present developments, challenges and trends. Int. J. Adv. Comput. Sci. Appl. 2017, 8. [CrossRef]

[38] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2021–2031.

[39] Barbara Rychalska, Dominika Basaj, and Przemyslaw Biecek. 2018. Are you tough enough? Framework for robustness validation of machine comprehension systems. CoRR, abs/1812.02205.

[40] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association of Computational Linguistics.

[41] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. arXiv preprint arXiv:1804.09541

[42] Mutabazi, E., Ni, J., Tang, G., & Cao, W. (2021). A review on medical textual question answering systems based on deep learning approaches. Applied Sciences, 11(12), 5456.

[43] Iftene, A. (2009). Textual Entailment (Ph. D. Thesis). Computer Science, University of Iasi, Iasi, Romania.

[44] Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., & Ngonga Ngomo, A. C. (2017). Survey on challenges of question answering in the semantic web. Semantic Web, 8(6), 895-920.

[45] Zhang, Y., Qian, S., Fang, Q., & Xu, C. (2019, October). Multi-modal knowledge-aware hierarchical attention network for explainable medical question answering. In Proceedings of the 27th ACM international conference on multimedia (pp. 1089-1097).

[46] Anastasios Nentidis, Anastasia Krithara, Konstantinos Bougiatiotis, Martin Krallinger, Carlos Rodriguez-Penagos, Marta Villegas, and Georgios Paliouras. 2020. Overview of BioASQ 2020: The Eighth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. In Experimental IR Meets Multilinguality, Multimodality, and Interaction, Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Hideo Joho, Christina Lioma, Carsten Eickhoff, Aurélie Névéol, Linda Cappellato, and Nicola Ferro (Eds.). Springer International Publishing, Cham, 194–214.

[47] Yoon, W., Lee, J., Kim, D., Jeong, M., & Kang, J. (2019, September). Pre-trained language model for biomedical question answering. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 727-740). Springer, Cham.

[48] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al., An overview of the bioasq large-scale biomedical semantic indexing and question answering competition, BMC bioinformatics 16 (2015) 1–28. doi:10.1186/s12859-015-0564-6.

[49] Hu, Z. (2019). Question Answering on SQuAD with BERT.

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018.

VITA AUCTORIS

NAME:	Malita Michael Dodti
PLACE OF BIRTH:	Maharashtra, India
EDUCATION:	University of Mumbai, Bachelor of Engineering in Computer Science, Mumbai, Maharashtra, India, 2018
	University of Windsor, M.Sc. Computer Science (with co-op), Windsor, ON, 2022