COVID-19 BASED DATASET KAGGLE CHALLENGE TASK SOLVING QUESTION AND ANSWER MODEL BASED ON NOVEL HIERARCHICAL MODEL ARCHITECTURE UTILIZING EXISTING STATE OF ART MODEL ALGORITHMS

Prashant Chadha
Student ID: 927706

Under the supervision of
SUVAJIT MUKHOPADHYAY

A thesis report submitted in fulfilment of the requirements of
Liverpool John Moores University for the degree of Masters in
Data Science

September 2020

ACKNOWLEDGMENT

I would like to thank my thesis supervisor, Mr Suvajit Mukhopadhyay, for his guidance, valuable and insightful feedback for the completion of the thesis. I would also like to thank DR. Manoj Jayabalan from Liverpool John Moores University for his continued support and guidance through weekly and one-on-one sessions throughout the duration of the program.
I am grateful to my parents, my wife, and my son for supporting and encouraging me throughout the duration of my study and preparing this interim report.

Thank you,
Prashant Chadha

ABSTRACT

The objective for my project is to solve Kaggle COVID-19 challenge Task query - "What is known about transmission, incubation, and environmental stability?" (Allen Institute For AI et al., 2020). Corona Virus has disrupted life world over and this Kaggle challenge aims to aid in research about this virus by providing answers to task-based questions related to COVID 19 like the one mentioned above.

This task will be solved with the help of deep learning based "Attention" concept. The Attention mechanism in Deep Learning is based off concept of directing focus and pay greater attention to certain factors when processing the data. It's one of the foremost methodologies utilized extensively in Q&A model design. The hierarchical model design will be created using two different stages: first stage will consist of document ranking or classification in order to filter relevant documents while second stage would be Bi-Directional Attention Flow (BIDAF) network to get the answers to the task. The dataset includes a sizeable number of documents and to shortlist top documents that would consist answers to the task requirements, documents will be ranked using two approaches: dot product of TF-IDF values of documents as well as those of task question. The other approach will include using BERT model to classify the documents as relevant and non-relevant document with respect to the task query. The shortlisted documents will then be fed into the Question and Answering model to get the required answer span. This model will be implemented using both, BERT again as Question and Answering model and another bidirectional attention flow-based model. The two model architectural results based on evaluation metrics will be compared and the one with better scores will be selected.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Table 1: List of abbreviations

Abbreviation	Expansion
ML	Machine Learning
AI	Artificial Intelligence
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
LSA	Latent Semantic Analysis
LDA	Latent Dirichlet allocation
Seq2Seq	Sequence to Sequence
DBQA	Document based Question and Answer
Q&A	Question and Answer
SQUAD	Stanford Question Answering Dataset
SIF	Smooth inverse function

Table 2: List of abbreviations

CHAPTER 1: INTRODUCTION

1.1 Background of the study

Corona Virus, first found in a patient in Wuhan-China at the end of 2019, has since spread in more than 180 countries across the globe and resulted in both, the loss of life and economy within these countries. It was declared pandemic by WHO in January and after disrupting lives in China, Italy and Spain at the end of last year and early part of 2020, it has caused massive loss of lives in countries such as US, India, Brazil, UK, Germany, France and stalled economic activities across the globe and continue to spread.
Scientists across the globe have been studying the virus characteristics in order to suggest ways to reduce the spread, aid in development of therapeutics and vaccines. Since the virus is highly contagious and lethal to certain population, made evident by its impact across the globe, it's important that scientists can focus on relevant information shared by fellow colleagues by their documented literature to find answers that can help in design of possible virus spread mitigating steps as well as development of cure. The Kaggle challenge, (Allen Institute For AI et al., 2020), is an initiative to encourage data scientist community to develop an algorithm that can aid scientists by learning from extensive literature and then giving relevant answers to the task query.
The selected task query delves with detailing the important characteristics of this Virus such as:

- Range of incubation periods for the disease in humans (and how this varies across age and health status) and how long individuals are contagious, even after recovery.
- Prevalence of asymptomatic shedding and transmission (e.g., particularly children).
- Seasonality of transmission.
- Physical science of the coronavirus (e.g., charge distribution, adhesion to hydrophilic/phobic surfaces, environmental survival to inform decontamination efforts for affected areas and provide information about viral shedding).
- Persistence and stability on a multitude of substrates and sources (e.g., nasal discharge, sputum, urine, faecal matter, blood).

Complete list of expected characteristics to report can be accessed at the website (Allen Institute For AI et al., 2020).
It's imperative that model designed is capable of learning from exhaustive number of detailed documents and can provide relevant answer span with respect to task query. It's for this requirement that bi-direction attention flow-based models are best suited for answering queries by retrieving relevant content from large corpus. The model architecture should also be able to shortlist or rank documents before retrieving the answer(s) as focusing on most relevant documents from large dataset corpus shared by Kaggle (Allen Institute For AI et al., 2020) would aid in improving the performance as well as accuracy of model. Therefore, the decision to design a hierarchical model to answer the task query.

1.2 Problem statement

The most extensively used concept now used in design of Q&A model is Attention Concept. As per the web article (Loye, 2019) on Attention mechanism: The Attention

mechanism in Deep Learning is based off concept of directing focus and pay greater attention to certain factors when processing the data.
In broad terms, Attention is one component of a network's architecture, and oversees managing and quantifying the interdependence:

- Between the input and output elements (General Attention)
- Within the input elements (Self-Attention)

Attention was originally introduced as a solution to address the main issue surrounding sequence to sequence (Seq2Seq) models or Encoder-Decoder model, and to great success.

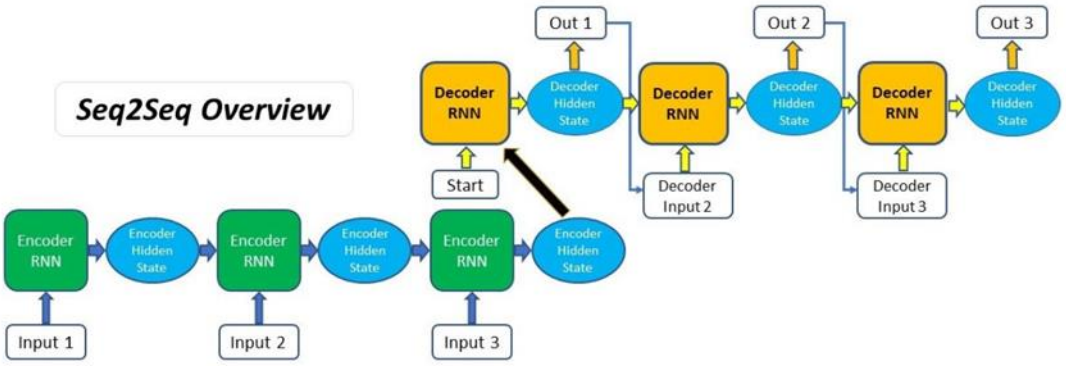

Figure 1: Seq2Seq Attention Process Flow in the Model **(Loye, 2019)**

The standard sequence-sequence model is generally unable to accurately process long input sequences, since only the last hidden state of the encoder RNN is used as the context vector for the decoder. On the other hand, the Attention Mechanism directly addresses this issue as it retains and utilizes all the hidden states of the input sequence during the decoding process. It does this by creating a unique mapping between each time step of the decoder output to all the encoder hidden states. This means that for each output that the decoder makes, it has access to the entire input sequence and can selectively pick out specific elements from that sequence to produce the output.
There are two types of Attention principle-based models:

- Bahdanau Attention
- Luong Attention

Word embeddings play major role in the NLP tasks-based models as they help model learn the semantic and syntactic relationships between the words. The research papers (Mikolov et al., 2013) and (Pennington et al., 2014) introduced novel word embeddings models, such as CBOW (Mikolov et al., 2013), Skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), that utilized individual context window and weighted co-occurrence word frequency concepts respectively. Models CBOW and Skip-gram have widely been used as word embeddings due to consistent performance and low complexity. The models focused on understanding the semantic and syntactic relationships between words within short context window and so would lose possible relationships between sentences separated by reasonable distance within given corpus. This would lead to performance issues in the case of question and answer models that are required to be trained over large corpus of data with answer spans consisting of sentences from different paragraphs.
GloVe model was designed with focus to overcome these shortcomings and the novel algorithm developed by authors helped in the model not only learning the semantic and

syntactic relationships between words but also the origin of these relationships and hence making the word embeddings more robust as well as versatile.

One of the research work, “A Unified Model for Document-Based Question Answering Based on Human-Like Reading Strategy” (Li et al., 2018), authors utilized Attention principle to design unified model that contains three major encoding layers that are consistent to different steps of the reading strategy, including the basic encoder, combined encoder and hierarchical encoder. Two levels of matching strategies are considered: the first is converting the whole source and target sentence into embedding vectors of latent semantic spaces respectively, and then calculating similarity score between them; the second is calculating the similarity score among all possible local positions of source and target sentences, and then summarizing the local scores into the final similarity score. Authors of this paper went with a novel approach of designing the model based on human-like reading strategy to tackle the DBQA problem, which could be conducted via the neural network.
The “summary” generation method utilized by authors, however, will not work in dataset considered for thesis project as answer need to be formulated from multiple documents and so the first and last sentences will not reflect the summary of complete data to be considered

“Attention Is All You Need” (Vaswani et al., 2017), research paper introduced concept of Transformer. Authors designed model architecture utilizing only attention mechanism without use of CNN or RNN network. They wanted to design a self-attention-based network that would avoid traditional RNN and CNN based model architecture bottlenecks at the time related to sequential computation and modelling of dependencies between text corpus without regard to their distance in the input or output sequences. Self-attention, mechanism that relates different portions of a single sequence in order to compute sequence representation, was achieved via implementation of a novel Transformer architecture consisting of stacked multi-layer Encoder and Decoder architecture. This paper helped in understanding concept of self-attention and its significance in learning relationship between texts separated over long distance in a sequence.
However, the proposed model architecture lacked sound strategy for short listing relevant documents from large corpus and so was not ideal for scenarios wherein dataset includes large number of documents or data sources.

Question and Answer model Framework capable of retrieving answers from long documents was introduced by “Coarse to Fine Question Answering for Long Documents” (Choi et al., 2017) research work. Authors developed a hierarchical network comprising of coarse, fast model for selecting relevant sentences and an expensive RNN based model to produce answer from the selected sentences. Sentence selection, treated as latent variable and trained together with answer generating model using re-inforcement learning, helped in selecting correct sentences as per context. Three types of mechanisms were considered for sentence selection model: Bag of words model, Chunking Bag of words model and Convolutional (CNN) model.
Model architecture showcased great performance on datasets with long documents but was only explored and studied for answer retrieval from a single document, therefore, different strategy is needed for dataset and task requirement of thesis project. Nonetheless, the sentence selection mechanisms utilized in this work were enlightening as this approach inspired the document shortlisting approach being considered in the current thesis project.

Most of the initial Question and Answer as well as other natural language processing (NLP) task-based models utilized single direction, i.e., either Left sequence learning or right sequence learning model. These worked on short sentences, but the contextual learning was never fully realized over long sequences. “Bi-Directional Attention Flow For Machine Comprehension” introduced Bi-Directional Attention Flow (BIDAF) network, a multi-stage hierarchical process that represents the context at different levels of granularity and uses bi-directional attention flow mechanism to obtain a query-aware context representation without early summarization. BIDAF includes character-level, word-level, and contextual embeddings, and uses bi-directional attention flow to obtain a query-aware context representation. Unlike few of other research works, author of this paper did not summarize the context paragraph into a fixed-size vector. Instead, the attention was computed for every time step, the attended vector at each time step, along with the representations from previous layers, can flow through to the subsequent modelling layer. This reduced the information loss caused by early summarization. Attention mechanism was implemented for both directions, query-to-context and context-to-query, which provided complimentary information to each other.
The model characteristics mentioned above made it ideal to be considered as one of the answer spans generating model in the hierarchical network to be designed to solve the task query. However, an information retrieval stage needs to be complimented with this model in order to efficiently generate answer span for given question from corpus consisting of large number of documents.

“BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” (Devlin et al., 2019) research work introduced the state of the art and the most versatile model yet, capable of solving multiple NLP use cases including Question and Answer task. It can be incorporated using transfer learning, pretraining on known large dataset and then fine-tuned on actual dataset. Authors introduced novel approach of using Masked language modelling (LML) that allowed deep bidirectional training by masking few random tokens and hence avoiding trivial responses from model. Model architecture consists of bidirectional multi-attention layers encoder and decoder architecture as introduced by research work (Vaswani et al., 2017).
Most of the recent and state of the art question and answer models proposed in research works such as (Nogueira and Cho, 2019), (Yang et al., 2018) and (Pappagari et al., 2019) have utilized BERT model and achieved state of the art results. The information retrieval stage will need to supplement BERT model as it performs best when answer span is required to be extracted from focused input data.

1.3 Aim and objectives

The main aim of this research is to develop a hierarchical network architecture capable of answering the Kaggle task query based on learning of content in extensive dataset. Therefore, goal of this research is to explore and demonstrate the performance of novel two stage model architecture options consisting of existing state of the art NLP task-based models.
The research objectives are formulated based on the aim of this study which is as follows:

- To develop robust and efficient question and answer model capable of learning content from exhaustive list of research documents and answer the task query
- To suggest the best performing hierarchical model network with respect to answers generated by model network options being considered

- To develop evaluation metric strategy most suited to judge the performance of model
- To identify appropriate granularity level to be considered for input sequence

1.4 Research questions

Following research questions are formulated as per the literature review and task solving strategy requirements:

- What's the most suited evaluation metrics to assess model performance taking into consideration answers are not explicitly available?
- What's the best granularity level (document, paragraph) to be considered as input to model?
- What's the ideal score to be considered while using TD-IDF values of query and data considered to have the answer?

1.5 Scope of the study

As per the fixed time frame schedule, below limitations are set on this research to ensure timely completion of work:

- The dataset considered is restricted to the pdf files shared over Kaggle website only.
- The objective of this research is restricted to solving the task query mentioned above. The task query will be represented as list of sub-questions, focusing on specific Corona virus details asked by the task query.
- Evaluation technique(s) formulation that suits for assessment of the proposed model design is also part of scope of this study.

1.6 Significance of the study

The research study would aid in development of model capable of answering Corona Virus related task query, given huge repository of research papers developed by scientists studying the Virus. This would help the scientists in getting required details about virus in a much more efficient and faster way. Thereby, enabling scientists suggest virus spread mitigation steps, development of therapeutic treatment.

1.7 Expected outcome

The expected outcome of this project is to develop hierarchical layers-based model capable of answering the task-based questions raised in Kaggle challenge (Allen Institute For AI et al., 2020), considering the extensive research papers as dataset shared in the same challenge.

1.8 Structure of the study

The chapter 1 focuses on the problem domain of this study. It provides details on COVID 19 related challenge undertaken and the "Attention" concept that will be utilized in design of the proposed model architectures. The "Problem statement" section details the concepts involved in question and answer models as well as some of the major issues faced in this

domain. The objectives set for this research are detailed in the “Aim and objectives” while ‘questions’ realised from literature review are mentioned in section ‘Research questions. The two sections, “Scope of the study” and “Significance of the study”, detail the scope of this research and its contribution to the field of COVID 19 study assistance as well as question and answer model architecture.

The chapter 2 is devoted to detail the literature reviewed carried out to learn the work done in the field of question and answer NLP task. This section consists of details related to research goal, methodology, results achieved, and gaps identified based on the learnings from literature review.
The Chapter 3 details the proposed methodology and dataset considered for developing solution for the challenge undertaken. This section provides brief description of the dataset and the pre-processing steps done for data preparation. The proposed model architectures and strategies developed to carry out the research and develop the solution are detailed in this section.
The Chapter 4 details the data analysis results as well as the description of the dataset. The various data transformations and processing methods are detailed, and the implementation of the proposed models and algorithms are discussed in detail within this chapter.
The Chapter 5 consists of results achieved by experiments conducted using the methodology detailed in previous chapters. The possible reasons for outcomes are also detailed based on the inferences.
The Chapter 6 details the conclusion made along with discussion on the research conducted. The future work that’s planned to be carried out after this project is also mentioned in this chapter. This last chapter is followed by References and appendices.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This section will focus on previous work on Question and Answer (Q&A) task solution with the help of AI or machine learning techniques. This section will review how the initial work became essential for current day state of the art model architectures, adoption of 'Attention' technique, extensive use of RNN/CNN neural networks and advent of current non-RNN based state of the art attention based Q&A model, like BERT, that have now become logical choice for solving many NLP tasks, including Q&A. Following sections will be divided as per literature being reviewed focussing on word embeddings, single and bi-directional attention flow models utilizing CNN/RNN based layers and lastly on work using non-CNN/RNN attention-based layers.

2.2 Word Embeddings: Key to understand syntactic and semantic word relationships

2.2.1 Efficient Estimation of Word Representations in Vector Space

In order to understand work on word embeddings, research paper "Efficient Estimation of Word Representations in Vector Space" (Mikolov et al., 2013) was referenced as it focussed on comparing the existing models at the time with two of the most popular and capable word embeddings models, CBOW and Skip gram, both of which designed by authors of this paper.

According to (Mikolov et al., 2013) the NLP systems and techniques at the time of their research work treated words as atomic units, i.e., no mention of similarity between words. The advantages of such simple techniques include ability of such models to train on large amount of data which resulted in them outperforming more complex models that could only be trained on considerably less data. N-gram is an example of one such model that can train on virtually all the available data.

The amount of relevant in domain data for automatic speech recognition, however, is limited as for most languages the corpora available for machine translation contain only a few billions of words or less. Since the performance of word embeddings model is governed mostly by the size of high-quality transcribed data, the simplistic models such as N-gram perform poorly in the case of insufficient data which's quite often the case.

During the years the processing power available has increased substantially and so complex models can be trained on substantial data, enabling them to outperform the models such as N-gram which require huge amounts of data to be capable performers.

The goal of this research though was to introduce novel techniques that can be utilized for learning high quality word vectors from huge data sets with less training time required as compared to the complex models available at the time. The data set comprised of billions of words, with millions of words in the vocabulary. The research team of paper seeked to design a new comprehensive test set for measuring both syntactic and semantic regularities and to show that such regularities can be learned with high accuracy.

Training time and accuracy dependence on the dimensionality of the word vectors as well as on the amount of training data was also analysed.

2.2.1.1 Research methodology

A popular model, proposed in (Bengio, 2003), uses neural network language model (NNLM) and consists of feedforward neural network with a linear projection layer and a non-linear hidden layer. It is used to learn jointly the word vector representations and a statistical language model.
The models proposed in this paper were inspired from another NNLM based architecture proposed in papers (Mikolov et al., n.d.). The strategy adopted in this model was to first learn word vectors using neural network with a single hidden layer. Word vectors were then used to train the NNLM model.
Authors selected this architecture as base as they observed that other initial available architectures were computationally expensive for training than the one selected. They focused on extending work on the architecture and focus on learning high quality word vectors, while keeping complexity as low as possible.
The focus of this research paper was on distributed representations of words learned by neural networks as such architectures perform significantly better than Latent Semantic Analysis (LSA) for preserving linear regularities among words. Latent Dirichlet Allocation (LDA), on the other hand, becomes computationally expensive on large data sets as observed by authors of this paper.
Authors defined the computational complexity, as per definition given in (Mikolov et al., 2011), as the number of parameters that need to be assessed to fully train the model. Training complexity thus, was defined for models considered for comparison as well as the novel models proposed as:
O = E x T x Q, where E: number of training epochs, T: number of words in the training set and Q: model architecture specific parameter.
Models proposed in this paper were trained using stochastic gradient descent and backpropagation.

2.2.1.1.1 Initial NNLM based models

As detailed in (Mikolov et al., 2013), the feedforward neural net language models available before the novel models proposed by this paper consists of input, projection, hidden and output layers. The input layer consists of 'N' previous words which are encoded using 1-of-V coding, V being size of the vocabulary.
Input layer is then projected to a projection layer 'P', dimensionality being N x D, using a projection matrix. Composition of projection layer is a relatively cheap operation as only 'N' inputs are active at any given time. NNLM architecture becomes complex for computation between propagation and hidden layer, as values on the projection layer are dense.
For example, if 'N' = 10, size of projection layer may be 500 to 2000, while hidden layer size 'H' would typically be 500 to 1000 units. Hidden layer is used to compute probability distribution over all the words in the vocabulary and so the output layer dimensionality becomes 'V'. The factors mentioned above results into huge computational complexity and is dominated by the term 'H x V'. Techniques proposed to avoid this term include using hierarchical versions of SoftMax, or, by using models that are not normalized during training. Such a binary tree representations of the vocabulary result in reduction in the number of output parameters to be evaluated to around 'lnV'.
Authors of this paper though used hierarchical SoftMax, wherein the vocabulary is represented as a Huffman binary tree. Huffman tress assign short binary codes to the

most frequent words, reducing further the number of output parameters that need to be evaluated.
The computational complexity of the NNLM model made researchers look for more optimized architectures for word embeddings model.
Recurrent Neural Net Language Model (RNNLM) was developed to overcome certain limitations of the feedforward NNLM, such as need to specify the context length (order of the model). The architecture does not include a projection layer and consists of input, hidden and output layers. Its capable of representing more complex patterns than the shallow neural networks and consists of recurrent matrix that connects the hidden layer to itself, using time delayed connections. This enables formation of short-term memory information from past represented by the hidden layer state, which gets updated based on the current input and the state of the hidden layer in the previous time step. Computational complexity of this architecture is again dominated by factor 'H x V', which can be reduced using hierarchical SoftMax. Such a model algorithm improved upon few issues in feedforward based NNLM and utilized inherent short-term memory functionality instead of complex projection layer to understand the relationships between the words. However, computational complexity was still not reduced.

2.2.1.1.2 Log-linear models

The focus while designing of such models first proposed in (Mikolov, n.d.), was to develop model architectures that learn distributed representation of words, while striving to minimize the computational complexity. The idea was to develop simpler models than neural networks-based models that could be trained on much larger data set efficiently, thereby increasing the accuracy while also reducing the training time. The research work (Mikolov et al., 2013) proposed two novel models "Continuous Bag-of-Words" and Continuous Skip-gram". Both models have become popular since their inception and have given state of the art results in generation of high-quality word vectors.
Continuous Bag-of-Words Model (CBOW): Model architecture proposed in (Mikolov et al., 2013), is similar to traditional feedforward NNLM. The complex hidden layer was removed, and the projection layer is shared for all words, and not just the projection matrix. Therefore, all words get projected into the same position and their vectors are averaged. The architecture is called "bag of words" as the order of words in the history does not influence the projection. Words that occur after the word to be predicted or words from future are also used while training the model. Model was built with a log-linear classifier with four future and four history words at the input. The training criterion considered was to correctly classify the current (middle) word based on learning from past and future words. The model architecture allows continuous distribution representation of the context and hence, model is aware of relationship between words separated over reasonable distances in a sentence. The training complexity is given by term:
$Q = N \times D \times D \times \log_2(V)$
It can be observed that computational complexity in this case is reduced as compared to traditional NNLM models.
Continuous Skip-gram model: Model is similar to CBOW as detailed in (Mikolov et al., 2013), however, focus is on maximising classification of a word based on another word in the same sentence, instead of predicting the current word based on the context learnt from neighbouring words on either side. The current word is used as an input to

log-linear classifier with a continuous projection layer. Words within certain range before and after the input word are then predicted. It was observed in that increasing () the range improves the quality of resulting word vectors; at the same time, it also increases the computational complexity. As distant words are less likely to be related to input word, they were given less weight by under sampling such words from the training samples. Training complexity is given by term:
$Q = C \times (D + D \times \log_2(V))$, where 'C' is the maximum distance of words. Words within the same sentence, 'R', to be predicted before and after the input word would lie with 1 and C and so the output would consist of 2xR words.

2.2.1.2 Research results and comparisons

This section consists of details related to the performance comparison conducted in this paper (Mikolov et al., 2013). It compared the novel models CBOW and Skip gram proposed in this paper with the state of the art NNLM models available at the time. Parallel training, wherein several models on top of a large scale distributed framework called 'DistBelief' are trained, was implemented for evaluation of models in (Mikolov et al., 2013). Mini-batch asynchronous gradient descent, with adaptive learning rate procedure – Adagrad, was used during the training procedure.
Simple algebraic operations performed with vector representations of words to find words like current word. For example, word similar to 'small', similar to the way biggest is similar to big, a vector can be computed X = vector ("biggest") – vector("big") + vector("small").
High dimensional word vectors were trained on a large amount of data, resulting vectors could then be used to answer subtle semantic relationships between words. Test set consisted of five types of semantic questions and nine types of syntactic questions, with total number of questions being 8869 semantic and 10675 syntactic questions. Also, only single token words were used, multi-word entities such as "New York" were not part of data set. Accuracy measure considered was stringent as synonyms were considered as mistakes.
Google news corpus was used for training the word vectors and it consists of 6 billion (B) tokens. Vocabulary size was restricted to 1 million (m) most frequent words. Models were trained on subsets of training data, with vocabulary restricted to most frequent 30 thousand (K) words. This way, choice of model architecture for getting best possible results as quickly as possible was achieved. Three training epochs with stochastic gradient descent and backpropagation were used, with starting learning rate of 0.025. Learning rate was decreased linearly so that it approached towards zero at the end of the last training epoch. Same training data set and same dimensionality of word vectors, 640, was considered for all models being compared. Although initially vocabulary was restricted to 30K. further experiments were conducted utilizing the full set of questions in test set as well.
CBOW architecture with different choice of word vector dimensionality and varying amount of training data was analysed. It was observed that at some point, adding more dimensions or more training data does not provide any further significant improvements. Further improvements achieved by increasing both vector dimensionality and the amount of training data. However, computational complexity increased at same rate with increase in training data or dimensionality of word vectors so there's a limit to the vector dimensionality and amount of data that can be considered.
The first comparison was conducted using same 640-word vector dimension.

Word vectors from RNN performed well mostly on syntactic questions. NNLM models though, performed significantly better than RNN.
CBOW performed better than NNLM (Mikolov et al., 2013)(authors of research paper developed their own NNLM as well apart from CBOW and Skip gram) on the syntactic tasks, and almost had similar results on the semantic tasks. Skip-gram performed better than NNLM but slightly worse than CBOW on the syntactic task. Overall, skip-gram performed the best considering both semantic and syntactic tasks within the test set. The training time was significantly less for novel models, CBOW and Skip gram, proposed by authors of this paper when compared to RNN and NNLM models.
Next, results were recorded by evaluating models trained using only one CPU and then performances were compared. In this case, the NNLM proposed in this paper performed best in the syntactic word-relationship test set, while skip-gram performed best overall.
It was also observed as part of another experiment that training a model on twice the data using one epoch gives better results than iterating over same data for three epochs.
The performance of the three models proposed in this paper (Mikolov et al., 2013) was also recorded on the basis of large scale training. The word vector size of CBOW and Skip-gram was considered as 1000, while for the NNLM it was kept at 100 due to higher computational complexity within the NNLM architecture. Skip-gram model was again, best overall with total accuracy being 65.6 and achieved best accuracy in semantic relationship test. CBOW had the best accuracy in the syntactic test and slightly less total accuracy when compared to Skip-gram. NNLM gave the worst performance and even though it consisted of word vectors of $1/10^{th}$ size of the vectors used in the other two, the training time for NNLM was considerably long when compared to CBOW and Skip-gram. This showcased that not only the novel models gave better word vectors but are also much faster to learn than NNLM and other word embeddings models available at the time of this research work.

2.2.1.3 Research gaps

The novel models proposed, CBOW and Skip-gram, in (Mikolov et al., 2013) definitely achieved state of the art results while being much more simpler and efficient when compared to existing models available at the time of research work. However, following are few gaps observed:

- Models were only trained on data set that consisted only single token words and no multi-word entities such as “New York”. This could result in performance degradation while using these models on data sets such as the one being used in my research work as it’s a corpus medical research work that consists of many multi-word entities.
- Performance of the two novel models proposed in this paper were great for custom test data set however, it was observed that performance of Skip-gram in the Microsoft sentence challenge was not the best and required to be complemented with RNN based Language model to achieve best scores.

2.2.2 GloVe: Global Vectors for Word Representation

“GloVe: Global Vectors for Word Representation” (Pennington et al., 2014) focused on developing a model capable of not only learning vector space representations of

words by capturing fine-grained semantic and syntactic regularities, but also understand the origins of these regularities. The proposed model in this paper leveraged statistical information as it was trained only on the non-zero elements in a word-word co-occurrence matrix. This was novel approach followed by authors of this paper as previous work to this research considered training proposed word embeddings model on data consisting of either entire sparse matrix, or, individual context windows within large corpus. Model was named GLoVe, global vectors, as it was developed with design philosophy to capture global corpus statistics directly.

2.2.2.1 Research methodology

The proposed model utilized the algorithm that enabled learning the meaning and relationship within words based on the ratio of their co-occurrence probabilities with various probe or context words. Authors developed novel weighted least squares-based cost function that accurately weighted the co-occurrences based on frequency, rather than weighing all of them equally. Model performance was evaluated on word analogy task (Mikolov et al., 2013), word similarity task (Luong et al., 2013) and the CONLL-2003 shared benchmark dataset for NER (Sang and De Meulder, 2003). The proposed model was trained on 2010 Wikipedia dump with 1 billion tokens, 2104 Wikipedia dump with 1.6 billion tokens, Gigaword 5 consisting of 4.3 billion tokens, combination of Gigaword 5 and Wikipedia 2014 which consisted of 6 billion tokens and 42 billion tokens of web data from Common Crawl.
The similarity score used as measure to judge performance was calculated by first, normalising each feature across the vocabulary and then calculating the cosine similarity. Authors then computed Spearman's rank correlation coefficient between the score and the human judgements.

2.2.2.2 Research results and comparisons

The proposed GLoVe model gave state of the art performance in all the mentioned tasks and outperformed the other benchmark setting models such as SVD, CBOW and Skip gram, often using corpus half the size compared to rest of models for training.
In the case of NER task, GLoVe outperformed all models such as SVD, CBOW, HPCA and HSMN on all the datasets considered, except CoNLL test set wherein HPCA model performed slightly better.
Authors observed that performance of the model was better on the syntactic subtask for small and asymmetric context windows. Semantic information, however, was captured with larger window sizes.
Model performance increased for syntactic sub-task with increase in the corpora used for training. However, semantic subtask performance relied more on the quality (accuracy and consistently updated) of dataset instead of the size of the dataset.

2.2.2.3 Research gaps

- The proposed model was not tested with scientific or medical based dataset
- Authors did not evaluate their proposed word embeddings model performance on NLP tasks such as question and answer task.

2.3 CNN/RNN Based Question and Answer Models

This section reviews previous work in the field of NLP task, Question and Answer, that consists of model architectures using CNN/RNN based layers.

2.3.1 A Unified Model for Document-Based Question Answering Based on Human-Like Reading Strategy

In one of the research papers, "A Unified Model for Document-Based Question Answering Based on Human-Like Reading Strategy" (Li et al., 2018), authors utilized Attention principle to design unified model that contains three major encoding layers that are consistent to different steps of the reading strategy, including the basic encoder, combined encoder and hierarchical encoder. They focused on designing the model that could imitate human-like reading strategy for document-based Question and Answering (DBQA) task. As stated by authors, in the field of sentence pairs matching, there have been various deep neural network models proposed.

2.3.1.1 Research methodology

Two levels of matching strategies are considered: the first is converting the whole source and target sentence into embedding vectors of latent semantic spaces respectively, and then calculating similarity score between them; the second is calculating the similarity score among all possible local positions of source and target sentences, and then summarizing the local scores into the final similarity score. Authors of this paper went with a novel approach of designing the model based on human-like reading strategy to tackle the DBQA problem, which could be conducted via the neural network. Word embeddings were based on word2vec pre-trained on the WiKiQA dataset.

The reading strategy, also called information retrieval stage, involved first making a "summary" or "title" embedded vector of document in the first stage. As per the second step, authors incorporated the hidden representation of the title into the question, posing a limitation to the understanding of it and making the meaning closer to the document.

Latent Semantic analysis (LSA) and Latent Dirichlet allocation (LDA) were used to get an overall summary of a document. These methods of getting the summary focused on first and last sentences in the document. Several methods, including deep learning models and simple computations, to combine both information was implemented by authors.

Thirdly, a hierarchical RNN structure was employed to obtain the document level representation, equipped with the new question's encoding vector as traditional RNN cannot capture the dependencies between number of sentences in the document.

The hierarchical encoder proposed in this paper (Li et al., 2018) consists of basic encoder, combined encoder and the bi-directional LSTM layer. Basic encoder and combined encoder were applied first for the sentences of the document separately. Basic encoder produced encoded vectors based on word embeddings representing sentences in the document and the question respectively. Modified version of LSTM/GRU was used to design basic encoder. The combined encoder was used in two stages, first to add document summary to the encoding of the question and second, to add the question's encoding vectors to the encoding of the document. The LSTM layer was then used to encode each sentence-question vector again as received from

the output of the combined encoder. The LSTM layer was also used to capture contextual features among sentences and made the understanding of a document more coherent.
Finally, a SoftMax layer was used to choose the answer sentence.

2.3.1.2 Research results and comparisons

The evaluation metrics were based on mean average precision (MAP) and mean reciprocal rank (MRR). The model was tested on English WikiQA dataset as well as Chinese DBQA dataset. It performed well on both datasets. First choice for document summary was observed to be the natural 'title' of a document. The concatenation computation between the question and the summary to update the representation of question gave the best performance.
The model architecture detailed in this paper helped develop insights about the hierarchical model design capable of filtering important content in the document and then use the same to answer respective query.

2.3.1.3 Research gaps

The "summary" generation method will not work in dataset considered for my thesis project as answer need to be formulated from multiple documents and so the information retrieval stage deployed in this paper of considering either the 'title' or the first and last sentences as document summary, will not reflect the summary of complete data to be considered in my data set. The information retrieval as well as the answer generation model architecture is more suited for single document as it will need lot of changes in the information retrieval stage.

2.3.2 Coarse to Fine Question Answering for Long Documents

Question and Answer model Framework capable of retrieving answers from long documents was introduced by "Coarse to Fine Question Answering for Long Documents" (Choi et al., 2017). Authors developed a hierarchical network comprising of coarse, fast model for selecting relevant sentences and an expensive RNN based model to produce answer from the selected sentences. Sentence selection, treated as latent variable and trained together with answer generating model using re-inforcement learning, helped in selecting correct sentences as per context.

2.3.2.1 Research methodology

Model algorithm consists proposed in (Choi et al., 2017) of two parts:

- Fast sentence selection model
- Costly answer generation model

Three types of mechanisms were considered for sentence selection model: Bag of words model, Chunking Bag of words model and Convolutional (CNN) model. Document summary was generated using the selected sentences using soft or deterministic attention as well as hard or stochastic attention. The summary was then fed to answer generating model. The answer generating model was designed using an encoder and decoder network based on Gated Recurrent Unit (GRU). The model

proved to be capable of generating answer that may not appear in the selected summary sentences verbatim.
Three types of learning approaches were used:

- Pipeline model learning, distant supervision wherein sentence selection model and answer generating model trained separately.
- Soft Attention learning, fully differentiable and optimized end-to-end with back propagation.
- Hard Attention approach: optimized with Reinforce algorithm

2.3.2.2 Research results and comparisons

Data set considered in this paper (Choi et al., 2017) include WIKIREADING, WIKIDATA and WIKISUGGEST. Google search was used to create question-answer pairs from pruned list of WIKIREADING based documents. Also, data was selected to make sure documents being considered consisted of more than 10 sentences so that short documents were not part of analysis. The models proposed were compared with following baseline models by authors for performance analysis:

- FIRST: model that selects first sentence in the document
- BASE: Re-implementation of the best model proposed in (Hewlett et al., 2015) , which consumes first 300 tokens
- ORACLE: selects the first sentence with answer string if it exits, otherwise considers the first sentence in the document.

Answer generation accuracy: BoW encoder used for sentence selection as it gave the fastest performance, out of the three mentioned options. Though, its accuracy was not the best for selecting sentences from WIKIREADING LONG dataset.
Reinforce or hard attention-based approach for learning detailed by authors was observed to perform at least three times faster than the BASE model.
Reinforce learning based model also outperformed Pipeline learning approach-based model. In fact, it outperformed all models being considered for comparison in the case of WIKIREADING LONG by authors, except ORACLE model (it had access to gold labels at the time of tests).
In the other two datasets, wherein answers are concentrated within first few sentences, BASE model performed the best. BASE model was observed to be advantageous in categorical based questions.
Reinforce or hard attention based training was observed to be the best amongst the three overall and it was also the most flexible as it allowed additional sentence to be included in the document summary during training procedure in case sentences selected do not result in accurate answer generation.
In terms of sentence selection for challenging dataset such as WIKIREADING LONG, complex models, CNN and CHUNK BoW, outperformed the simpler BoW.
The models proposed in this paper, therefore, gave state of the art results when compared with few of the existing models at the time on same dataset and different learning strategies as well as sentence selection model performances were compared.
Model architecture showcased great performance on datasets with long documents but was only explored and studied for answer retrieval from a single document.

2.3.2.4 Research gaps

- The proposed model architecture in this paper (Hewlett et al., 2015) focused on retrieving answers from single document. However, my thesis project work requires for model to learn from exhaustive list of documents and so again the information retrieval method used is not applicable for my project.
- The proposed model architecture gave good results in the three data sets considered but was not always the best compared to other architectures at the time, so better and more efficient model architectures are now available.

2.3.3 Bi-Directional Attention Flow for Machine Comprehension

"Bi-Directional Attention Flow For Machine Comprehension" (Seo et al., 2016) introduced Bi-Directional Attention Flow (BIDAF) network that utilized a hierarchical a multi-stage architecture for modelling the representations of the context paragraph using bi-directional attention flow. BIDAF consists of character-level, word-level and contextual embeddings and thus, represented context at different levels of granularity.

2.3.3.1 Research methodology

The attention mechanism deployed by authors of this paper (Seo et al., 2016) improved upon the previously popular attention paradigms as follows:

- The attention, instead of being used to summarize the context paragraph into a fixed size vector, was computed for every time step. The attention vector along with representations from previous layers then flows through to the subsequent modelling layer. This reduced the information loss caused by early summarization.
- Memory-less attention mechanism wherein the attention at each time step is a function of only the query and the context paragraph at the current time step was used. It did not directly depend on the attention at the previous time step and authors hypothesized that such a mechanism would force the attention layer to focus only on learning the attention between the query and context.
- Attention mechanisms were used in both directions, query-to-context and context-to-query, and thereby captured the complimentary information given by attention over respective directions.

The model architecture that will be used consists of following layers:

- Character Embedding Layer maps each word to a vector space using character-level CNNs.
- Word Embedding Layer maps each word to a vector space using a pre-trained word embedding model.
- Contextual Embedding Layer utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context.
- Attention Flow Layer couples the query and context vectors and produces a set of query aware feature vectors for each word in the context.
- Modelling Layer employs a Recurrent Neural Network to scan the context.

- Output Layer provides an answer to the query. The answer would be span within the context of the query.

2.3.3.2 Research results and comparisons

Model was evaluated on SQUAD (Rajpurkar et al., 2016) dataset. It consists of 100K questions and answer generated was a span within the context. Evaluation metrics used include Exact Match (EM) and F1 score, which measured the weighted average of the precision and recall rate at character level. Dataset was divided into 90K training and 10K test data sets. Results achieved were comparable with other leading model architectures at the time of this research (Seo et al., 2016), when single model iteration performance was compared. However, in the case of ensemble model setting benchmark topping scores were achieved with EM score of 73.3 and F1 score of 81.1, outperforming the existing approaches at the time of proposal of this novel approach. It was also observed that word-level embedding was better at representing the semantics of each word, while char-level embedding helped in understanding of rare or out-of-vocab (OOV) words.

2.3.3.3 Research gaps

- The proposed model (Seo et al., 2016) doesn't include information retrieval stage required to filter most relevant documents in scenarios with large corpus of documents. Therefore, this model requires to be complimented with separate module that focusses on ranking of most relevant documents.
- Model has only been evaluated on SQUAD dataset and so performance on dataset that's more scientific oriented is not certain.
- The pre-trained model is not tested on any other dataset and so the more optimum use of the model will be after training it on the considered dataset, which is not possible within the scope of this project.

2.4 Non-CNN/RNN Based Question and Answer Models

2.4.1 Attention Is All You Need

"Attention Is All You Need" (Vaswani et al., 2017), research paper introduced concept of Transformer. Authors designed model architecture utilizing only attention mechanism without use of CNN or RNN network. They wanted to design a self-attention-based network that would avoid traditional RNN and CNN based model architecture bottlenecks at the time related to sequential computation and modelling of dependencies between text corpus without regard to their distance in the input or output sequences. Self-attention, mechanism that relates different portions of a single sequence in order to compute sequence representation, was achieved via implementation of a novel Transformer architecture consisting of stacked multi-layer Encoder and Decoder architecture. Authors also examined the performance of different types of attention based mathematical algorithms such as scaled dot-product attention, consisting of either additive or dot-product attention, and Multi-head attention.

2.4.1.1 Research methodology

The proposed model (Vaswani et al., 2017) had Encoder consisting of stack of six identical layers. Each layer consisting of two sub layers: multi-head attention layers (8) and a feet forward neural network. Both sub layers surrounded by addition and normalization layer steps.
Decoder design consisted of similar structure but has an additional Masked Multi-head attention layer. One of the multi-head attention layers is common between encoder and decoder. Linear and Soft-max layer follows the decoder, with output sequence generated by Soft-max layer.
Since no CNN/RNN layers were used, authors used positional encodings and added positional encodings to the input embeddings at the bottom of the encoder and decoder stacks. This injected some information about the relative or absolute position of the tokens in the sequence and had the same dimension as that of the learned embeddings utilized in this model.

2.4.1.2 Research results and comparisons

Standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs and WMT 2014 English-French dataset consisting of 36M sentences were used for training the model proposed (Vaswani et al., 2017).
Utilizing multi-head attention layer architecture state of the art results comparable with existing language translation models at the time were achieved on same datasets at much faster pace. This paper helped in understanding concept of self-attention and its significance in learning relationship between texts separated over long distance in a sequence.

2.4.1.3 Research gaps

- The proposed model (Vaswani et al., 2017) only focussed on language translation tasks so the SoftMax layer as well as the layers used in the model not tweaked or tested on question-answer tasks.
- The proposed model doesn't include information retrieval stage required to filter most relevant documents in scenarios with large corpus of documents. Therefore, this model requires to be complimented with separate module that focusses on ranking of most relevant documents.

2.4.2 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (Devlin et al., 2019) research work introduced the state of the art and the most versatile model yet, capable of solving multiple NLP use cases including Question and Answer task. It can be incorporated using transfer learning, pretraining on known large dataset and then fine-tuned on actual dataset. Authors introduced novel approach of using Masked language modelling (LML) that allowed deep bidirectional training by masking few random tokens and hence avoiding trivial responses from model. Utilizes bidirectional transformer network wherein encoder reads the entire sequence of words at once, allowing the model to learn the context of a word based on all its surrounding words.

2.4.2.1 Research methodology

Multi-layer bi-directional transformer based on the original implementation (Vaswani et al., 2017). Two models proposed in this paper, Bert-Base consisted of 12 layers, hidden network layer of the size of 768 and 12 attention heads. The total parameters to be trained were 110M. Second model was BERT-Large consisting of 24 layers, hidden network layer of the size of 1024 and 16 attention heads. The total parameters to be trained were 340M.
Word Piece embeddings with 30K vocabulary was used to create word embeddings. Two special tokens, [CLS] and [SEP] were used. Token [CLS] used to store the averaged output probabilities from SoftMax layer representing the classifier result, or, start and end points in the case of answer span generation, given an input question and context. The [SEP] token separated sentences within the context or input sequences.
Model was pre-trained using two unsupervised tasks, Masked Language Model (LM) and Next Sentence Prediction (NSP)
Masked LM consisted of masking some percentage of the input tokens at random, predicting these tokens to train a deep bi-directional representation. Fifteen percent of all Wordpiece tokens in each input sequence were masked at random and only predicted the masked words rather than reconstructing the entire input. The [MASK] token was used 80% of the time for masking, a random token was selected 10% of the time and 10% of the time original token was selected. This was done to avoid proposed bi-directional attention-based model to indirectly "see" the word during input of the sequence and hence, trivially predict the word.
The NSP task involved model being pre-trained for a binarized next sentence prediction task. Sentences 'A' and 'B' for each pre-training example were selected in a way such that 50% of the time 'B' was the actual next sentence that followed sentence 'A' while, the other 50% of the time it was a random sentence from the corpus. This helped the model understand relationships between the sentences.
The model was then fine-tuned or feature-tuned on various NLP tasks and performance was compared with existing state of the art models at the time of this research.
Fine-tuning involved selecting the final SoftMax layer as per the NLP task at hand, while, feature-tuning involved extracting only the features learned from BERT model and using them to train another model.

2.4.2.2 Research results and comparisons

The proposed Models, both Base and Large variants, in this paper (Devlin et al., 2019) was fine-tuned on 11 NLP tasks and gave state of the art results on all of them and has since set benchmark scores yet to be surpassed. The datasets of interest were SQUAD v1 and SQUAD v2 as both are some of the best datasets available for validating performance of model on Question and answer task. Both models performed exceptionally well as the EM and F1 scores recorded were state of the art and set the benchmark scores.
Authors also noted that increasing the size of the model improves the performance on small scale tasks as well, provided model has been sufficiently pre-trained. Authors also concluded that proposed BERT model performed well on feature-tuned tasks as well making BERT a versatile model architecture.

2.4.2.3 Research gaps

The proposed model (Devlin et al., 2019) doesn't include information retrieval stage required to filter most relevant documents in scenarios with large corpus of documents. Therefore, this model requires to be complimented with separate module that focusses on ranking of most relevant documents so that limited documents would be required to be learned by final Question and Answer model.

2.4.3 Passage Re-Ranking with Bert

"Passage Re-Ranking with Bert" (Nogueira and Cho, 2019) implemented BERT (Devlin et al., 2019) for query-based passage re-ranking. Authors of this paper focused on second stage, i.e., passage re-ranking or information retrieval, of a typical question and answer model. This stage involves scoring of documents part of corpus, which forms the context from which answer needs to be identified for a given query. The documents are then re-ranked by a computationally intensive method based on the scores

2.4.3.1 Research methodology

The purpose of the re-ranker model proposed in this paper (Nogueira and Cho, 2019) was to estimate a score representing the relevance of a candidate passage to given query. BERT was used as the re-ranker with inputs consisting of query as sentence 'A' and the passage text as sentence 'B'. Query was truncated to be of size of at most 64 tokens and the passage text length was fixed to size such that the concatenation of query, passage and separator tokens had maximum length of 512 tokens. BERT-Large was used as binary classifier, with [CLS] vector retrieved as output of the final layer representing the probability of passage being relevant.
Probabilities of each passage were independently evaluated and the final list of passage by ranking with respect to probabilities (scores) was obtained.

2.4.3.2 Research results and comparisons

The proposed model (Nogueira and Cho, 2019) was trained on two passage ranking datasets, MS MARCO and TREC-CAR. MS-MARCO contains approximately 400M tuples of query, relevant and non-relevant passages. Development set contained approximately 6900 queries, each paired with the top 1000 passages retrieved using BM25 from the MS MARCO corpus. Evaluation set with approximately 6800 queries and top 1000 retrieved passages without relevance annotations was also used.
TRE-CAR (Dietz et al., 2012) consists of input query formed from concatenation of Wikipedia article title and the title of one of its sections. The corpus consisted of all the English Wikipedia paragraphs, except the abstracts. Authors evaluated model performance using automatic annotations in this case, providing relevance scores for all possible query-passage pairs. Also, model was trained only on half of Wikipedia used by TREC-CAR's training set in order to avoid leak of test data into training.
State of the art results were achieved despite using only a fraction of data available for training. Proposed model performance surpassed previous benchmark setters for this task by large margins on both the datasets.

2.4.3.3 Research gaps

The proposed model (Nogueira and Cho, 2019) focused on datasets that consists of query-passage pairs only but was not evaluated for dataset including corpus of large number of documents. The model seems ideal for information retrieval stage but needs to be modified for ranking documents consisting of number of passages within them, rather than just ranking passages with respect to query.

2.4.4 Anserini-Bert-SQUAD based Kaggle submission

The Kaggle Challenge submission, "https://www.kaggle.com/dirktheeng/anserini-bert-SQUAD-for-semantic-corpus-search" (Dirk, 2020), implemented Anserini + BERT architecture (Yang et al., 2019a) to create a question and answer model architecture capable of answering questions mentioned in the Kaggle challenge task based on the available corpus of documents.

2.4.4.1 Research methodology

Model architecture consists of Semantic search engine based on Anserini and BERT-SQUAD to produce best answer. Anserini search engine is built on top of Apache Lucene and it was used to search through available literature based on combination of query and "key words". BERT was then used to return relevant passages that were received from Anserini, considering semantic relationship between the query and passages. The passages were then ranked based on query using Google's "Universal Sentence Encoder", which produces a similarity matrix that mathmatically represents the semantic similarity of all the answers to that of the question. BART model, released by hugging face transformers, was then used to create summary using top few ranked passages

2.4.4.2 Research results and comparisons

The proposed model architecture in this article (Dirk, 2020) created accurate and crisp summaries based on the ranked passages and answered the requirements raised by questions.

2.4.4.3 Research gaps

The proposed model (Dirk, 2020) architecture consists of search engine to retrieve data from COVID-19 based database which worked fine for author's use case but not necessarily required when corpus of documents as data source is available. Its more efficient to develop information retrieval system that can simply rank or classify relevant documents out of the available corpus, which can then be used to retrieve desired answer with respect to query.

2.4.5 End to End Open Domain Question Answering with BERT Serini

"End to End Open Domain Question Answering with BERT Serini" (Yang et al., 2019a) work utilized BERT with open source Anserini information retrieval toolkit. Focus of this paper was to identify answers from large corpus of Wikipedia articles

with an end-to-end solution approach, consisting of both, a sound information retrieval solution and state of the art answer span generation system based on BERT.

2.4.5.1 Research methodology

The proposed model architecture (Yang et al., 2019a) consists of two main modules, single stage Anserini retriever and the BERT reader. The retriever stage was responsible for selecting segments of text within the available corpus that's relevant to the query. These segments were then passed to the BERT reader for identification of answer span with respect to the query.
Authors experimented with different granularities of text at the time of indexing. Such experiments included article (5.08M Wikipedia articles indexed), paragraph (29.5M paragraphs indexed) and sentence (79.5M sentences indexed) level granularities. Authors retrieved 'K' text segments (either paragraph or sentence) using the question as "bag of words" query. BERT reader was then used to select the best text span within the 'K' text spans and scores are assigned to each span. The reader and retriever scores were then combined via linear interpolation. The resulting equation consists of hyperparameter that's tuneable. The passage is then selected based on the top score.

2.4.5.2 Research results and comparisons

Evaluation metrics used included Exact Match (EM) and F1 score (at the token level). Results were compared for the performances of the model architecture (Yang et al., 2019a) using different granularities for indexing. Recall score, considering fraction of questions for which correct answer appeared in any retrieved segment. Initially, parameter 'K' was set as 5 for article level retrieval, 'K' = 29 for paragraph retrieval and 'K' = 78 for sentence retrieval. The article level granularity gave the worst performance while paragraph level gave the best performance at 'K' = 100. The possible reason hypothesized by authors for these results was the fact that article level granularity results in long passages that contain lot of non-relevant sentences that acted as distraction to the BERT reader. The sentence level granularity gave performance that lied between the two and though BERT reader wasn't required to read lot of non-relevant data within a given input sequence containing possible relevant data as well, the input sequence lacked required context for the reader to identify the exact answer span.
The granularity considered therefore, was paragraph level for tuning the model towards best performance as well as comparison with other state of the art models. Model performance was observed to increase with increase in 'K' till value of 100, when it attained best performance scores and outscored the other state of the art question and answer models.

2.4.5.3 Research gaps

There was room for improvement observed for BERT reader (Yang et al., 2019a) and the passage scoring mechanism as even though high percentage of Recall score was achieved, the top 'K' Exact Match and top '1' Exact Match scores were lower and reasonable gap existed between the scores.

2.4.6 Hierarchical transformers for long document classification

"Hierarchical transformers for long document classification" (Pappagari et al., 2019) focused on developing a BERT model based architecture that improves on BERT'S limitation related to requirement for input sequence size work to be just few hundred words. BERT has been stated of the art model for majority of NLP tasks however, authors mentioned in their paper that certain NLP tasks that require understanding context within data based on human conversations are not suited to standard BERT model. Such data could consist of long sequences that would exceed the maximum input sequence limitation of BERT model. Authors of this paper introduced novel approach consisting of BERT model and then either a single recurrent layer or another transformer layer followed by SoftMax activation layer. These two approaches were named as Recurrence over BERT (RoBERT) and Transformer over BERT (ToBERT).

2.4.6.1 Research methodology

The proposed model architecture (Pappagari et al., 2019) consists of two main modules, first BERT used to obtain representations of short sequences formed out of original input sequence and then used either a Recurrent LSTM network or another transformer to get the actual classification.
In the case of RoBERT, the input sequence was split into segments of fixed size with overlap and fed into the BERT model. The segment level representations achieved from BERT were then stacked into a sequence and then fed into a small LSTM layer. The output achieved from LSTM layer served as document embedding. The final predictions were achieved using final layer structure that consisted of two fully connected layers with 'ReLu' and SoftMax as activation functions respectively. Similar methodology used in the case of ToBERT but with small transformer model implementation after BERT instead of LSTM layer.

2.4.6.2 Research results and comparisons

Datasets used for this research (Pappagari et al., 2019) included CSAT dataset for CSAT prediction, 20 newsgroups for topic identification task, consisting of written text and Fisher phase 1 corpus for topic identification task, consisting of spoken transcripts. Results were recorded for performance of the model architecture based on features extracted from both, pre-trained BERT as well as fine-tuned BERT models. The performance in the case of utilizing only the pre-trained BERT led to subpar results when compared to fine-tuned BERT model for feature extraction. This was consistent with results based on evaluation over all the three datasets. ToBERT implementation performed better in for all the three datasets and was substantially better in the case Fisher dataset.
The proposed model setup performance was also compared with model strategies consisting use of either averaging of predictions extracted from the fine-tuned BERT, or, considering the most frequent prediction. It was observed that the performance of proposed model architecture of using another model, LSTM or transformer, for classification after the fine-tuned BERT provided substantial improvements as the size

of documents increased, with ToBERT consisting of fine-tuned BERT giving best results for Fisher dataset. The performance of ToBERT was comparable to the other two mentioned strategies in the case of other two datasets.
The performance of both, RoBERT and ToBERT, was state of the art for all the datasets when compared to other models. ToBERT outperformed RoBERT for classification tasks in the case of the three datasets and gave best performance in the case of Fisher database, setting up new benchmark at the time when compared to other models.

2.4.6.3 Research gaps

The proposed model (Pappagari et al., 2019) focused on datasets that consists of speech based texts only and was utilized for classification tasks only. The work from authors provided good insight on ways to tackle large documents however, there are better model architectures available for question and answer tasks with focus on information retrieval and answer span generation with respect to given query.

2.4.7 SQUAD: 100,000+ Questions for Machine Comprehension of Text

"SQUAD: 100,000+ Questions for Machine Comprehension of Text" (Rajpurkar et al., 2016) focused on developing large and high quality reading comprehension dataset based on exhaustive list of Wikipedia articles. The developed dataset consists of questions that were posed by Crowdworkers, with answers being a segment of text or span from the articles. Authors used logistic regression model to assess the difficulty of the proposed SQUAD dataset in terms of learning the data by ML/AI model.

2.4.7.1 Research methodology

Authors collected data for this research (Rajpurkar et al., 2016) in three stages consisting of the following:

- Curating passages
- Crowdsourcing question-answer pairs generation on these passages
- Extracting additional answers

Passage creation: Authors used project Nayuki' Wikipedia's internal PageRanks to obtain top
1000 articles of English Wikipedia and sampled at random, 536 articles. The resulting dataset was 23,215 paragraphs from these articles based on diverse topics, without tables, images, figures and shorter than 500 characters length paragraphs. The dataset was partitioned randomly into training set (80%), development set (10%) and test set (10%).
Question-answer pair generation: Crowdworkers were employed to create questions over Daemo platform (Group, 2015), with Amazon Mechanical Turk as backend. They were tasked to review paragraphs and create up to 5 questions per paragraph and answers for the same had to be highlighted within the respective paragraphs. Sample paragraph, valid and invalid questions were given as an example for guidance to Crowdworkers.
Additional answers collection: Two additional answers for each question in the development and test sets were obtained to make the evaluation strategy robust.

Crowdworker for this task was shown questions along with paragraphs of an article and was then asked to select shortest span within paragraph that answered the question. Question was required to be submitted without answer in case it was not answerable by span in the given paragraph.
Dataset evaluation model: Logistic regression model was then used by authors to evaluate the quality of proposed dataset in terms of diversity, complexity and validity for training models. The performance of logistic model used by authors was compared with selected baseline models, while using the proposed SQUAD dataset for question and answer task. Authors computed unigram/bigram overlap between sentence containing candidate answer and the question. All candidates within with maximum overlap were considered and best one was selected using the sliding-window based approach (Richardson et al., 2013). Authors also used distance based extension (Richardson et al., 2013) but used only the sentence containing the candidate answer instead of the entire passage.
Authors built around 180 million discretized features based on the proposed SQUAD training data and created questions based on the same. Most of the features lexicalized features or dependency tree path features. Features created helped the model select correct sentences, choose correct spans, learn common answer spans and understand the correct answer types.

2.4.7.2 Research results and comparisons

Two metrics, Exact match and F1 score, were used for evaluation Exact match measured the percentage of predictions that matched any one of the three ground truth answers exactly. The F1 score measured the average overlap between the prediction and ground truth answer. The model (Rajpurkar et al., 2016) performance was compared with three baseline models and human performance. The results achieved by proposed logistic regression model were better than the three baseline model algorithms considered (random guess, sliding window and sliding window plus edit distance), but were just average with F1 being 51 and Exact match score of 40.4%. These scores were significantly inferior to human performance recorded for the same task.
Most important features observed were lexicalized and dependency tree path.
The model performed best on predicting answers based on dates and other numerics, cases where answers were mostly single tokens and only few possible candidates were shortlisted.

2.4.7.3 Research gaps

- The dataset developed (Rajpurkar et al., 2016) by authors is one of the most extensive and widely used in research since its inception for model training and evaluation. However, it doesn't focus specially on medical or scientific fields.
- The model used based on logistic regression gave only average performance and has been surpassed by recent state of the art model designs that are capable of learning both syntactic and semantic meaning in a much better way. This results into much better and consistent performance by neural networks-based model design approaches and hence, are more suitable for question and answer tasks over large corpus and long answer spans.

2.4.8 BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

The proposed model by authors of this paper (Lewis et al., 2020), BART, is a denoising autoencoder for pretraining sequence-to-sequence models. It uses a standard transformer based neural machine translation architecture and is based on generalising BERT (due to the bidirectional encoder), GPT (with left-to-right decoder) and few of the other more recent pretraining schemes.

2.4.8.1 Research methodology

The proposed BART model architecture consists of a sequence-to-sequence model (Vaswani et al., 2017) with a bidirectional encoder (BERT) and a left-to-right autoregressive decoder – (GPT). The corrupted document is encoded with the bidirectional model and then the likelihood of the original document is calculated with the autoregressive decoder, for a typical machine translation task.
Authors used six layers in the encoder and decoder for the base model, while, twelve layers were used in each of them for the large model version. The architecture is similar to BERT (Devlin et al., 2019) but with following differences:

- Each layer of the decoder additionally performs cross-attention over the final hidden layer of the decoder.
- BERT utilizes an additional feed-forward network before word prediction, which is not the case in BART model.
- BART consists of around 10% more parameters than the equivalently sized BERT model/

BART model was trained by corrupting documents and then optimizing a reconstruction loss, which is the cross-entropy between the decoder's output and the original document.

The text transformation techniques tried by the authors during the pre-training of the proposed model are as follows:

- **Token masking:** Random tokens are sampled and replaced with [MASK] elements.
- **Token deletion:** Random tokens are deleted from the input. In contrast to token masking, the model must decide which positions are missing inputs.
- **Text infilling:** This is a novel approach by the authors of this paper for text transformation. Several text spans are sampled with span lengths drawn from a Poisson distribution and each span is then replaced with a single [MASK] token. The objective of text infilling is to teach the model to predict the number of missing tokens from a span.

- **Sentence permutation:** A document is divided into sentences based on full stops, and these sentences are shuffled in a random order.
- **Document rotation:** Token is chosen uniformly at random, and the document is rotated so that it begins with the selected token. This process trains the model to identify the start of the document.

Following fine-tune tasks were used to evaluate the model performance with some of the existing state of the art models:

- **Sequence-classification tasks**: The same input is fed into the encoder and decoder, and the final hidden state of the final decoder token is fed into new multi-class linear classifier. This approach is similar to the 'CLS' token in BERT; however, the authors added the additional token to the end so that representation for the token in the decoder can attend to decoder states from the complete input.
- **Token classification tasks**: A complete document is fed into the encoder and decoder of the model and the top hidden state of the decoder is used as a representation of each word. This representation in turn, is used to classify the token. An example of such a task is the answer endpoint classification for SQUAD dataset.
- **Sequence generation tasks**: The proposed model BART can be fine-tuned for sequence generation tasks such as abstractive question-answering and summarization. In this task, the encoder input is the input sequence and the decoder generates outputs autoregressively.
- **Machine translation**: The BART's encoder embedding layer is replaced with a new randomly initialized encoder. The model in then trained end-to-end in order to train the new encoder to map the foreign words into an input that BART can de-noise to English. The new encoder can use a separate vocabulary from the original BART model.

2.4.8.2 Research results and comparisons

Following approaches/models were used to compare the performance of BART on various NLP tasks:

- **Language model:** The authors trained a left-to-right transformer language model, similar to (Radford et al., 2019)
- **Permuted language model:** As in (Yang et al., 2019b), $1/6^{th}$ of the tokens are sampled and these are generated in a random order autoregressively in this model implementation.
- **Masked language model:** Following the BERT (Devlin et al., 2019), 15% of the tokens are replaced with [MASK] tokens and then such a model is trained to independently predict the original tokens.

- **Multitask masked language model:** Such a masked language model was designed similar to UniLM (Dong et al., 2019) and trained with additional self-attention masks. Self-attention masks were selected randomly in with following proportions: $1/6^{th}$ left-to-right, $1/6^{th}$ right-to-left, $1/3^{rd}$ unmasked, $1/3^{rd}$ with the first 50% of tokens unmasked and left-to-right mask for the remainder.
- **Masked seq-to-seq:** This model's implementation is based on MASS (Song et al., 2019) wherein, the authors masked a span containing 50% of tokens and trained the sequence to sequence model to predict the tokens.

The tasks considered for evaluation were treated either as standard sequence-to-sequence problem, or, the source is added as prefix to the target in the decoder with a loss only on the target part of the sequence. The first approach favours the proposed BART models and latter is suited for other models.
The tasks and datasets considered for evaluation are as follows:

- **SQUAD** (Rajpurkar et al., 2016) is a dataset based on Wikipedia paragraphs used for an extractive question answering task.
- **MNLI** (Williams et al., 2018) is a bitext classification task to predict whether one sentence entails another.
- **ELI5** (Fan et al., 2019) is a dataset used for long-form abstractive question answering task.
- **XSum** (Narayan et al., 2018a) is a news summarization dataset used for abstractive summarization task.
- **CONVAI2** (Dinan et al., 2019) is a dialogue response generation task, conditioned on context and a persona.
- **CNN/DM** (Hermann et al., 2015) is a news summarization dataset used for summarization task.

Following observations were made during evaluation of base BART models and the other models considered for comparison:

- The effectiveness of pre-training methods is highly dependent on the task. For example, a simple language model achieved the best performance on ELI5 dataset-based task but struggled in the SQUAD based task.
- Token deletion, masking and self-attention masks-based pre-training objectives performed much better than the rotating documents or permuting sentences-based objectives. Deletion based objective in fact appeared to outperform masking on generation tasks.
- The left-to-right pre-training improves model performance in generation-based tasks.
- In the case of SQUAD task, bidirectional encoders-based models perform much better than left-to-right decoders-based models. The proposed BART

model variants achieved similar performance with only half the number of bidirectional layers when compared to traditional bidirectional encoders.

- Architectural improvements such as relative position embeddings or segment level recurrence are equally important as the pre-training objective in achieving high performance from the model.
- BART achieved the most consistently strong performance over all the tasks except the ELI5 task. The other language-based models considered during evaluation performed better on this task which showed that BART is less effective when the output in loosely constrained by the input.

The detailed results can be accessed from table-1 in (Lewis et al., 2020)

The proposed BART model was also evaluated on large-scale corpus. The authors trained BART using the same scale dataset as the RoBERTa (Liu et al., 2019). The large variant of BART consisted of 12 layers in each of the encoder and decoder, and a hidden size of 1024. The large variant of the proposed model was evaluated on following tasks:

- **Discriminative tasks:** The performance of BART model is compared with large variant of models such as BERT, UniLM, XLNet and RoBERTa on SQUAD and GLUE based tasks. The performance of the BART model was state of the art and comparable to benchmark setting model RoBERTa on most tasks. This showed that BART's improvements on generation tasks did not come at the expense of classification task performance. The table-2 in (Lewis et al., 2020) can be referred for more details.
- **Generation tasks:** The proposed large variant BART model's performance is also evaluated on tasks such as Summarization, Dialogue and Abstractive QA. The CNN/DailyMail and XSum datasets were used for evaluation on Summarization task. The models considered for comparison were Lead-3, PTGEN (See et al., 2017), PTGEN + COV (See et al., 2017), UniLM, BERTSUMABS (Liu and Lapata, 2020) and BERTSUMEXTABS (Liu and Lapata, 2020). BART outperformed all the models considered for comparison and gave benchmark setting scores in both, CNN/DailyMail and XSum, datasets. The table-3 in (Lewis et al., 2020) can be referred for more details.

  In the case of Dialogue task, BART again outperformed previous work on two automated metrics on the CONVAI2 dataset. The models considered for comparison were Seq2Seq + Attention and Best system. The table-4 in (Lewis et al., 2020) can be referred for more details.
- **Translation:** The BART model was also evaluated on WMT16 Romanian-English task (Sennrich et al., 2016). The authors used a 6-layer transformer source encoder to map Romanian into a representation that BART was able to de-noise into English. The model performance was compared with baseline transformer architecture (Vaswani et al., 2017).

The performance was compared with both fixed BART and tuned BART variants. The tuned BART performed better than the Baseline and fixed BART models, but the author's approach was less effective without back-translation data and prone to overfitting. The table-6 in (Lewis et al., 2020) can be referred for more details.

2.4.8.3 Research gaps

- The proposed BART model performed well and gave state-of-the art results in most NLP tasks such as extractive QA, Summarization, Classification but did not perform well in abstractive QA task on ELI5 dataset, when compared to language models.
- The author's approach while using the proposed BART model in translation task was observed to be less effective and was prone to overfitting. Additional regularization techniques can be explored to improve the performance.
- The performance of the BART model was only tested on datasets that focussed on content available or generated in media. Therefore, model's understanding of dataset focussed on medical topics such as COVID19 may not be great which could impact the model's performance as I will be using a pre-trained model.

2.4.9 Fine-tune BERT for Extractive Summarization

BERTSUM, the model proposed in this paper (Liu, 2019), is a simple variant of the BERT (Devlin et al., 2019) model that was designed for extractive summarization. The authors of this paper selected this model as its pre-trained on a huge dataset and consists of a powerful architecture, capable of learning complex features that would result in boost in the performance of extractive summarization.

2.4.9.1 Research methodology

The model is designed in a way that each sentence in a document, consisting of several sentences is assigned a label {0,1} indicating whether the sentence need to be considered for summary. It's based on assumption that summary sentences represent the most important content of the document.
The BERT model is required to output the representation of each sentence for the extractive summarization task. Following are the two updates performed on the original sentence encoding and embedding for the use of BERT in extractive summarization task:

- **Encoding multiple sentences:** The token [CLS] was added for each sentence (instead of just at the beginning of first sentence as in vanilla BERT) and a

[SEP] token was added after each sentence. The multiple [CLS] symbols added here were used to get features for the sentences ascending the symbol.

- **Internal segment embeddings:** The authors used internal segment embeddings to distinguish multiple sentences within a document. For example, a sentence - '$sent_i$' will be assigned a segment embedding 'E_A' or 'E_B' conditioned on 'i' being odd or even. Therefore, a sequence of sentences [$sent_1$, $sent_2$, $sent_3$, $sent_4$] were assigned [E_A, E_B, E_A, E_B]. The vector $T_{i,}$ which is the vector of the i^{th} [CLS] symbol from the top BERT layer, is used as the representation for '$sent_i$'.

Fine-tuning with summarization layers: Several summarization-specific layers were stacked on top of the BERT outputs that captured the document-level features from the sentence vectors obtained from BERT for the extractive summaries. A final score was predicted for each sentence and the loss of the whole model was the binary classification entropy of the score against a gold label.

Following are the type of summarization layers that were used on top of the BERT output:

- **Simple classifier:** In this case, a linear layer on the BERT outputs was added and a sigmoid function was used to get the predicted score.
- **Inter-sentence transformer:** In such a summarization layer, more transformer layers are applied only on sentence representations. These transformer layers extract document-level features by focussing on summarization tasks from the BERT outputs. It was observed by the authors as per performance evaluation that transformer with two layers performed the best. The final output layer is again a sigmoid classifier.
- **Recurrent neural network:** The authors of this paper also applied LSTM layer over the BERT outputs to learn summarization specific tasks. The final output layer again was a sigmoid classifier.

2.4.9.2 Research results and comparisons

The 'bert-base-uncased' version of the BERT model was used and both BERT and summarization layers were jointly fine-tuned. The model was first used to obtain the score for each sentence and the same were then ranked by the scores in descending order. The top three sentences were then selected as summary.

Trigram blocking: This technique was used to reduce redundancy within the selected sentences for summary. Given a selected summary 'S' and a candidate sentence 'C', the sentence 'C' is skipped if a trigram overlapping exists between 'C' and 'S'.

Dataset details

The evaluation of the proposed BERTSUM model's performance was done using the two benchmark datasets, the CNN/DailyMail news highlights dataset and the NewYork Times Annotated corpus.
The CNN/DailyMail dataset consists of news articles and associated highlights. The training, validation and testing split for CNN documents was (90,266/1,220/1,093) and (196,961/12,148/10,397) for DailyMail documents.
The NYT dataset consists of 110,540 articles with abstractive summaries. The split for training and test datasets was (100,834/9706) respectively and around 4000 examples from the training dataset were selected as the validation set.
The documents with summaries shorter than 50 words were removed from the raw dataset. The filtered dataset included 3,452 test examples.
The authors used greedy algorithm to generate an oracle summary for each document in the case of both datasets. This was done as both datasets consisted of abstractive gold summaries that are not well suited for training extractive summarization models. The algorithm selected sentences that maximise the ROUGE scores as the oracle sentences.

Evaluation strategy and results

The model and its variants proposed were compared with existing state-of-the art summarising models at the time of this paper such as LEAD, REFRESH (Narayan et al., 2018b), NEUSUM (Zhou et al., 2018), PGN (See et al., 2017) and DCA (Çelikyilmaz et al., 2018) in the case of CNN/DailyMail dataset. The proposed model variants, vanilla BERT, BERTSUM + Classifier, BERTSUM + Transformer and BERTSUM + LSTM, outperformed all the existing summarization models mentioned above for the CNN/DailyMail dataset. BERTSUM with transformer summarization layer achieved the best performance on all the three metrics considered during model evaluation (R-1, R-2 and R-L). The BERTSUM + LSTM did not improve the performance on the summarization task when compared to the BERT + Classifier model. Both interval segment and trigram blocking increased the base model performance and summarization results respectively. For more details, refer table-1 in (Liu, 2019).
In the case of NYT datasets, the predicted summaries were truncated to the lengths of gold summaries and the summarization quality was evaluated with ROUGE Recall. The models used for comparison were First-K words, Full (Durrett et al., 2016) and Deep reinforced (Paulus et al., 2017). The BERTSUM + Classifier achieved the state-of-the art results and outperformed all the three mentioned models for this dataset. Refer table-3 in (Liu, 2019) for more details.
The proposed BERTSUM model architecture in this paper with inter-sentence transformer layers achieved the best performance at the time on the two benchmark datasets.

2.4.8.3 Research gaps

- The performance of the BERTSUM model was only tested on datasets that focussed on content available or generated in media. Therefore, model's understanding of dataset focussed on medical topics such as COVID19 may not be great which could impact the model's performance as I will be using a pre-trained model.
- The performance of the BERTSUM model in the summarization task considered has been bested by more recent model, BART (Lewis et al., 2020).

2.5 Summary

The literature review was carried out with a purpose to first understand the word embeddings concept that plays big part in training models for NLP tasks. The authors of research paper (Mikolov et al., 2013) did a robust research by exploring existing word embeddings approach at the time and developed novel models, CBOW and Skip-gram, that outperformed existing models not only in terms of getting better evaluation metrics scores but were also faster to train and required less machine resources.

However, models were only trained on data set that consisted only single token words and no multi-word entities such as "New York" which can lead to performance issues for models that need to be trained on datasets that consists of important multi-word entities and medical research papers.

Also, the novel algorithm used in the two models was designed using individual context window approach, which works best in the case of question and answer models that need to return answer spans consisting of short number of sentences that lie close to each other within the corpus. This again would lead to performance issues in the case of datasets that may require models to learn semantic relationships from sentences that may be separated and lie in different paragraphs within the corpus. The dataset considered for my research consists of similar structure and so this attribute of the two models may not give the best results.

The research paper (Pennington et al., 2014) details word embeddings model developed by authors using novel approach that was designed to improve upon shortcomings of both individual context window approach as well as word co-occurrence frequency approach. Authors developed algorithm that utilized weighted word co-occurrence frequency probabilities that helped the model learn not only the semantic and syntactic relationships within words but also the origins of such relationships. The proposed model outperformed all state-of-the-art word embeddings models at the time in most of the semantic and syntactic evaluation tests for the selected NLP tasks. However, the model was not evaluated on learning ability of science and medical domain-based dataset and neither was it evaluated for performance in question and answer task.

The focus for literature review was then shifted to understand the question and answer model architecture itself. It was learnt that the "Attention" concept essentially forms the basis of such model architectures and it's implemented using encoder-decoder based transformer architecture. Also, the architecture consists of two stages, 'Information retrieval' stage that's used to extract relevant data with respect to a question from given corpus and then 'Reader' stage that selects the answer span relative to the question from the extracted data. I focused on more recent literature that

utilized either the CNN/RNN based neural layers to implement “Attention” concept to develop state of the art question and answer model, or, non-CNN/RNN based feed-forward neural layers to develop such models.
The “Information retrieval” stage was developed using algorithms such as TF-IDF and Bag of Words (BoW) in some of the initial research work but the latest implementations have utilized search engines such as Anserini or state of the art and versatile BERT model. The ‘Reader’ stage was implemented using either CNN/RNN in some of previous state of the art model implementations or, feed-forward based neural network layers that have been utilized in the more recent top performing models, including the current industry leading BERT model.
The literature read related to CNN/RNN based models showcased some of the best implementations of ‘Attention’ principle and state of the art performances were achieved. The architecture proposed in (Seo et al., 2016) implemented ‘Attention’ at various granularities within corpus and gave state of the art performance in comparison to other CNN/RNN models at the time. However, inherent issues within CNN/RNN based layers such as model complexity and inconsistent performance in terms of learning semantic relationship within words separated over long distances resulted in limitations with regards to improving the accuracy of such models.
The authors of research paper (Vaswani et al., 2017) developed novel model architecture that utilized stacked multiple feed-forward based layers in both the encoder and decoder to implement bi-directional ‘Attention’ and showcased great performance on question and answer task. This work was also the inspiration behind the state of the art and versatile model BERT (Devlin et al., 2019). Most of the recent question and answer research work has implemented BERT in either, or both, the “Information retrieval” and “Reader” stage and achieved benchmark setting results, outperforming previously proposed state of the art models.
The literature related to extractive summarization was also explored as the summarization task is part of my proposed model architecture. The authors of paper (Liu, 2019) updated the vanilla BERT model and included summarization layers which helped in much improved performance in summarization tasks. The proposed model BERTSUM in this paper outperformed all the existing models at the time in extractive summarization task and included novel sentence encoding techniques such as Internal segment encoding and performance optimization technique such as Trigram blocking.
The model, BART, proposed in (Lewis et al., 2020) again utilized original BERT bidirectional encoder and updated it by including left-to-right decoder (GPT). The inclusion of autoregressive decoder improved the vanilla BERT model architecture performance in tasks such as extractive summarization and translation.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This section details the approach followed for developing model architecture that will be used to solve the task undertaken.
Following are the procedural steps carried out so far with the goal to develop sound strategy to develop the required solution:

- Read and understand the Kaggle challenge (Allen Institute For AI et al., 2020) put up on Kaggle platform to encourage model architecture submissions that can assist medical doctors and scientists in their research on COVID 19.
- Collect data available on the task website (Allen Institute For AI et al., 2020) and analyse it for understanding as well as for design of pre-processing functions
- Read literature and view online content to learn about work done in the field of NLP tasks such as Word Embeddings and Question and Answer task. The knowledge gained by following this process has assisted in developing couple of model architecture options that seem novel and ideal to solve the requirements of this task.
- Collect reference answers from trusted sites related to the set of questions being considered for this project
- Develop the proposed model and evaluate answers generated with respect to the reference answers to analyse the performance of proposed model architecture in solving the task undertaken for this thesis.

The following section will detail the model architecture as well as the high-level data description that has been learned and designed so far following the above-mentioned steps.

3.2 Data collection and usage

- Dataset (Allen Institute For AI et al., 2020) consists of pdf and xml documents in jason format. The scope of this research will be restricted on the pdf documents in jason format.
- Apart from the above mentioned documents, there are 23 questions based on three tasks mentioned in the Kaggle competition (Allen Institute For AI et al., 2020). These questions will be used to get answers from the proposed model architecture
- The evaluation of the answers generated by model will be based on semantic similarity between the generated answers and reference answers. The reference answers have been extracted from medical sites and following are their citations: (Davies et al., 2020), (UKRI, 2020a), (University of California, 2020), (WHO, 2020), (ACOG, 2020), (Harvard Health Publishing, 2020), (UKRI, 2020b), (Forster et al., 2020), (Huang et al., 2020).

3.3 Data Pre-processing and Transformation

Data that needs to be processed consists of research work on Corona Virus by authors of respective papers. The "body_text" and "abstract" dictionaries within each document consist of research data and summary of the same respectively and as part of data pre-processing, two lists consisting of "body_text" and "abstract" data respectively from all documents within one of the folders (there are four folders as mentioned above) are created. Therefore, in all eight such lists would be created. The name of respective documents is saved in another list in same order.
The concept behind creating such lists is that it becomes easier to track the respective folder containing the documents relevant to task queries as per results of the document shortlisting models.
Once the list is created, following are the steps considered as part of data transformation before being used as input to the respective models:

Pre-process data (Query as well as data within the list) for the TD-IDF based document selection model:

- Converted all characters within the query as well as respective document data within the list to lower case.
- Used "RegEX" function to make sure only alpha numeric characters or valid web links exist in data using custom filter condition
- Removed multiple spaces between sentences by replacing such whitespaces with single whitespace.
- Used "TreebankWordTokenizer" function, available within python package, to split sentences into individual words or tokens.
- Removed stop words (most frequent words that are non-essential for query-sentence contextual relationship task) from the above list with tokens using python function, "stopwords".
- Used "stemmer" function to stem the tokens.
- Joined the resulting tokens with single space to form sentences.

Pre-process data (Query as well as data within the list) for the BERT model used for answer span generation from the abstracts of the top 10 documents shortlisted with the TF-IDF based document selection model:

- The BERT model has its own tokenization function that was used to encode and tokenize the context and the question.
- The BERT's tokenization function tokenizes the question and context pair together and utilizes its own 'Wordpiece' embeddings to encode the pair.
- The context-question encoded pair also consists of two special tokens, [CLS] and [SEP], that the BERT model utilizes for span generation and separator between the context and question.

3.4 Model Architecture

The proposed model architecture was changed from the one proposed in the proposal as well as the interim report. The changes were made as the CNN based Bi-Directional Attention Flow (BIDAF) network (Seo et al., 2016) model that was originally considered for implementation did not seem ideal for a pre-trained model use case

scenario. The high-level architectural diagram of hierarchical network now being proposed to provide answer to the task queries is as follows:

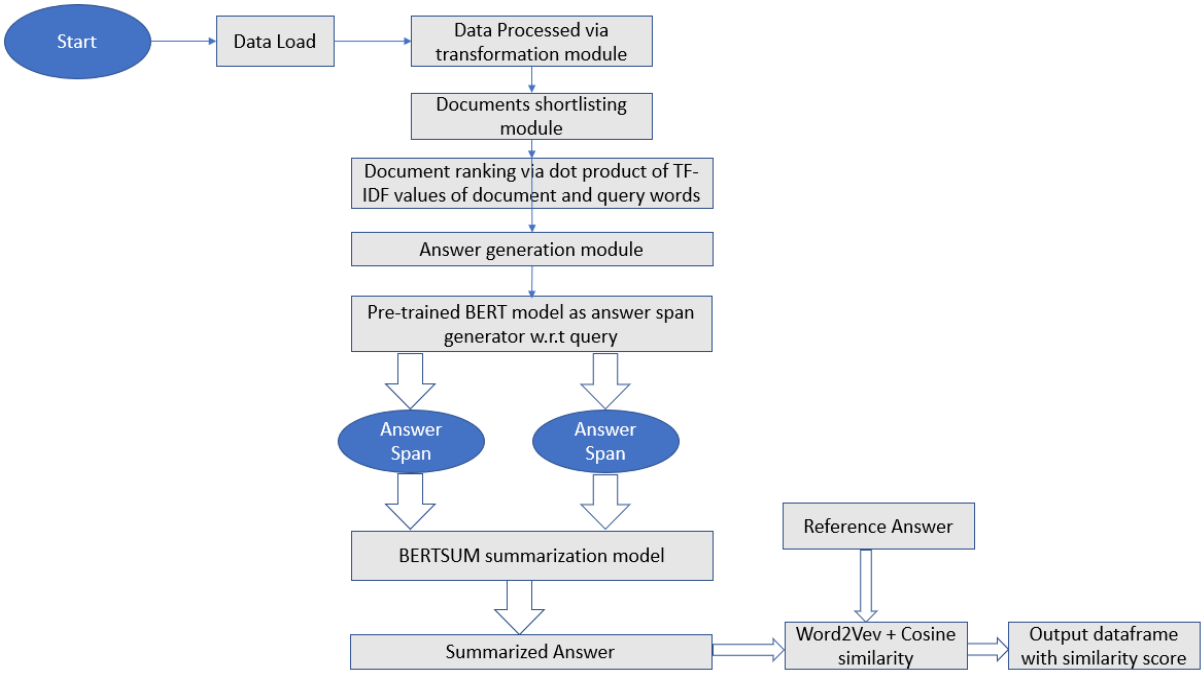

Figure 2: Hierarchical model process flow diagram

3.4.1 TF-IDF document ranking model

Term frequency–inverse document frequency (TFIDF) is a numerical statistic intended to reflect the importance of a word with respect to a document in a collection or corpus. It has often been used as a weighting factor in searches of information retrieval, text mining, and user modeling. The TF–IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. This helps to adjust for the fact that generally some words appear more frequently in text corpus.
The transformed data and query are transformed into respective TF-IDF word-importance value matrix using NLTK package function for TF-IDF vector value generation, “TfidfVectorizer”.
These vector values are then converted into python dataframe, wherein rows represent TF-IDF values of words in respective documents and columns are words that exist in the documents within the list.
Cosine product between the TF-IDF values in the dataframe and query TF-IDF values is taken. The resulting score for each document in the dataframe is then taken by summing up the values over each row, representing the combined relevance of words within respective documents with respect to the query. Top 10 documents are then selected for consideration during the answer generation stage. The apt number of documents based on ideal cut-off score is one of the research questions.

Model Selection criteria

The model algorithm, TD-IDF and cosine product between respective vectors, was selected for document ranking and filtering as its simple, fast and a proven algorithm

for search and filtering tasks in the field of NLP. The algorithm does lack ability to distinguish the meaning of sentences due to order of the words however, the algorithm has been considered for ranking the documents and not the answer generation so the assumption is that deep semantic understanding is not required at this stage of the model architecture.

3.4.2 BERT model

Bidirectional encoder representations from transformers, BERT (Devlin et al., 2019) model uses bidirectional attention to extract deep relationship between the query-sentence pair with help of novel approach, Masked Language Model (MLM). Since its inception, model has proven to give state of the art results in various NLP tasks such as language translation, sentiment analysis, named entity recognition (NER) as well as question and answer (Q&A) classification and answer span generation with respect to query.
The model architecture is based on self-attention based multi-layer encoder-decoder architecture (Vaswani et al., 2017). BERTBase (L=12, H=768, A=12, Total Parameters=110M) (Devlin et al., 2019), with L being the number of layers or transformer blocks, H is the hidden vector size and A is the number of attention heads, will be first used and based on performance as well as time, BERTLarge variant may be used.

Tokenization and Special Tokens

The word tokenization used by BERT, "Wordpiece", will be used for tokenizing sequence consisting of both query and passage (document or paragraph within document based on granularity). "Wordpiece" embeddings consist of vocabulary of 30,000 tokens. In case a token in the input is not present, the word is broken into pieces/individual characters and are mapped with ID from the embeddings library. The maximum acceptable sequence length is 512 tokens and it's a hyperparameter that can be tuned. In case the sequence is longer, its truncated and if the size of a given input sequence is less than the set size, its padded using special token [PAD].
[CLS] and [SEP] are the two special tokens utilized by BERT. [CLS] is the first token of input sequence, comprising of query and document data stream (document or paragraph). [SEP] token separates the query and the data sequence from documents. The role of [CLS] token is dependent on the final output expected from model.

In case of this project, [CLS] token will be used as follows:

- The start and stop indicators to highlight the answer span in the case of BERT used as answer span generation with respect to query.

3.4.2.1 Model training

Pre-trained BERT model, trained on SQUAD dataset (Rajpurkar et al., 2016), will be used. Fine tuning will be performed on the project dataset. Following are the important and novel pre-training approaches utilized for BERT model:

3.4.2.1.1 Masked LM (MLM)

BERT utilizes deep bidirectional attention with help of novel masked LM technique. This involves masking of some percentage of tokens and then let model predict these tokens for it to learn deep bidirectional representation within sequence. The strategy followed is as proposed by author (Devlin et al., 2019), i.e., 15% of all Wordpiece tokens in each sequence are masked at random and only these masked tokens are predicted.
Within this 15% masked tokens, 80% tokens are masked, 10% tokens are replaced by random tokens while rest of 10% are left as original tokens. The 15% tokens are selected randomly.

3.4.2.1.2 Next Sentence Prediction (NSP)

This process helps in model understanding the relationship between two sentences. As part of this pre-training task, two sentences, sentence A and B respectively are selected from text corpus. 50% of the time, sentence B is the actual next sentence to following sentence A and is labelled as “IsNext”, while, for rest of 50% cases, sentence B is a random sentence from the corpus and is labelled as “NotNext”.

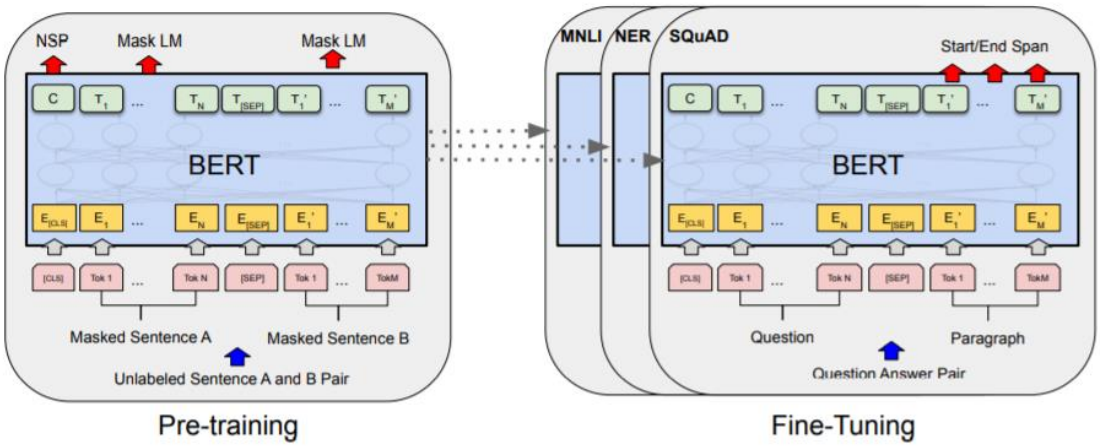

***Figure 3: BERT pre-training and Fine-Tuning process* (Devlin et al., 2019)**

3.4.2.2 BERT Fine-tuning

BERT fine tuning is a straightforward process and is carried out over the dataset being considered for NLP task such as Question-Answer, language translation. The special token [CLS] is tuned to generate the task specific output.
during the answer span generation task.
In the case of Question-Answer task, the self-attention mechanism utilized by BERT to encode both the query and passage concatenated tokens simultaneously and hence, enables bidirectional cross section attention between the query and passage. This helps in getting the required answer span with the help of [CLS] tokens.

3.4.2.3 Model variant used and selection criteria

Since the dataset considered for my project consists of limited questions and respective reference answers that are not enough for even fine-tuning, the BERT model package fine-tuned on SQUAD database for question-answer task will be used.

The BERT model was selected as it has given state of the art results in most of the NLP tasks and performed the best in question-answer when compared to other existing top performing models on datasets such as SQUAD. It's also proven to give excellent results when used in transfer learning mode, as the BERT model fine-tuned on SQUAD database has proven to give good results even on different datasets (Dirk, 2020). Therefore, this model was the best fit for my project because dataset does not consist enough question-reference answer pairs for running fine-tuning exercise over it.

3.4.3 BERTSUM-BERT for extractive summarization

BERTSUM model consists of the BERT (Devlin et al., 2019) model enhanced with summarization layers that enable the model to perform well in summarization task. The model is designed in a way that each sentence in a document, consisting of several sentences is assigned a label {0,1} indicating whether the sentence need to be considered for summary. It's based on assumption that summary sentences represent the most important content of the document.
The BERT model is required to generate the representation of each sentence for the extractive summarization task. Following are the two updates performed on the original sentence encoding and embedding for the use of BERT in extractive summarization task:

- **Encoding multiple sentences:** The token [CLS] was added for each sentence (instead of just at the beginning of first sentence as in vanilla BERT) and a [SEP] token was added after each sentence. The multiple [CLS] symbols added here were used to get features for the sentences ascending the symbol.
- **Internal segment embeddings:** The authors used internal segment embeddings to distinguish multiple sentences within a document. For example, a sentence - '$sent_i$' will be assigned a segment embedding 'E_A' or 'E_B' conditioned on 'i' being odd or even. Therefore, a sequence of sentences [$sent_1$, $sent_2$, $sent_3$, $sent_4$] were assigned [E_A, E_B, E_A, E_B]. The vector $T_{i,}$ which is the vector of the i^{th} [CLS] symbol from the top BERT layer, is used as the representation for '$sent_i$'.

The high-level diagram of BERTSUM model is as follows:

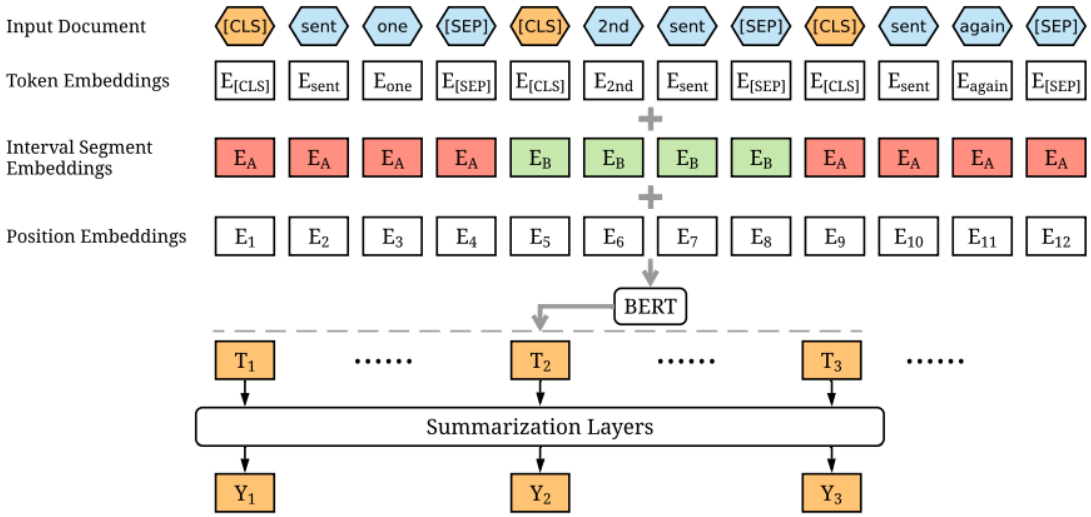

Figure 4: BERTSUM high level diagram (Liu, 2019)

3.4.3.1 Model training

BERTSUM model consists of BERT model along with summarization layers that are added to fine-tune the BERT model for better performance in the summarization task. The BERT model used here is the pre-trained model, that's trained using the techniques, Masked LM (LML) and Next Sentence Prediction (NSP), as detailed in the 'BERT model' section above.

3.4.3.2 Fine-tuning with summarization layers:

Several summarization-specific layers were stacked on top of the BERT outputs that captured the document-level features from the sentence vectors obtained from BERT for the extractive summaries. A final score was predicted for each sentence and the loss of the whole model was the binary classification entropy of the score against a gold label.

Following are the type of summarization layers that were used on top of the BERT output:

- **Simple classifier:** In this case, a linear layer on the BERT outputs was added and a sigmoid function was used to get the predicted score.
- **Inter-sentence transformer:** In such a summarization layer, more transformer layers are applied only on sentence representations. These transformer layers extract document-level features by focussing on summarization tasks from the BERT outputs. It was observed by the authors as per performance evaluation that transformer with two layers performed the best. The final output layer is again a sigmoid classifier.

- **Recurrent neural network:** The authors of this paper also applied LSTM layer over the BERT outputs to learn summarization specific tasks. The final output layer again was a sigmoid classifier.

The BERTSUM model used consists of additional transformer as summarization layer.

3.4.3.3 Model variant used and selection criteria

Since the dataset considered for my project do not consist of any reference summaries required for fine-tuning, the BERTSUM model package (Huggingface transformers) fine-tuned on CNN/DailyMail and NYT benchmark datasets was used.

The BERTSUM model was selected as it has given state of the art results in the summarization task on benchmark datasets, CNN/DailyMail and NYT. The BART summarization model was also considered however, the model did not work in the nimblebox environment due to possible compatibility issues.
The BERTSUM model has also proven to give excellent results when used in transfer learning mode, as the BERTSUM model fine-tuned on the datasets mentioned above has proven to give good results even on different datasets. Therefore, this model was the best fit for my project because the dataset does not consist summaries for running fine-tuning exercise over them.

3.4.4 Word2Vec word embeddings model to estimate summary-reference answer similarity

A Word2Vec (Mikolov et al., 2013) word embeddings model along with cosine similarity function is used to estimate the similarity between the summaries generated by proposed
model architecture and the reference answer. This will also help in gaging the performance of the proposed model in creating summaries with respect to given questions. Once the embedding vectors have been generated by the Word2Vec model instance, cosine similarity is then calculated, and respective scores are assigned to the dataframe.

The Word2Vec is a shallow two-layer neural network that's trained to reconstruct linguistic contexts of words. It produces a vector space for large corpus of words, with each unique word in the corpus being assigned a corresponding vector in the space. It comes in two variants that share similar algorithm and are detailed below:

Continuous Bag-of-Words Model (CBOW): The Word2Vec model architecture proposed in (Mikolov et al., 2013), is similar to traditional feedforward NNLM. The complex hidden layer was removed, and the projection layer is shared for all words, and not just the projection matrix. Therefore, all words get projected into the same position and their vectors are averaged. The architecture is called "bag of words" as the order of words in the history does not influence the projection. Words that occur after the word to be predicted or words from future are also used while training the

model. Model was built with a log-linear classifier with four future and four history words at the input. The training criterion considered was to correctly classify the current (middle) word based on learning from past and future words. The model architecture allows continuous distribution representation of the context and hence, model is aware of relationship between words separated over reasonable distances in a sentence. The training complexity is given by term:
$Q = N \times D \times D \times \log_2(V)$
It can be observed that computational complexity in this case is reduced as compared to traditional NNLM models.

Continuous Skip-gram model: Model is similar to CBOW as detailed in (Mikolov et al., 2013), however, focus is on maximising classification of a word based on another word in the same sentence, instead of predicting the current word based on the context learnt from neighbouring words on either side. The current word is used as an input to log-linear classifier with a continuous projection layer. Words within certain range before and after the input word are then predicted. It was observed in that increasing () the range improves the quality of resulting word vectors; at the same time, it also increases the computational complexity. As distant words are less likely to be related to input word, they were given less weight by under sampling such words from the training samples. Training complexity is given by term:
$Q = C \times (D + D \times \log_2(V))$, where 'C' is the maximum distance of words. Words within the same sentence, 'R', to be predicted before and after the input word would lie with 1 and C and so the output would consist of 2xR words.

Following is the high-level diagram depicting the working of the above two variants of Word2Vec model:

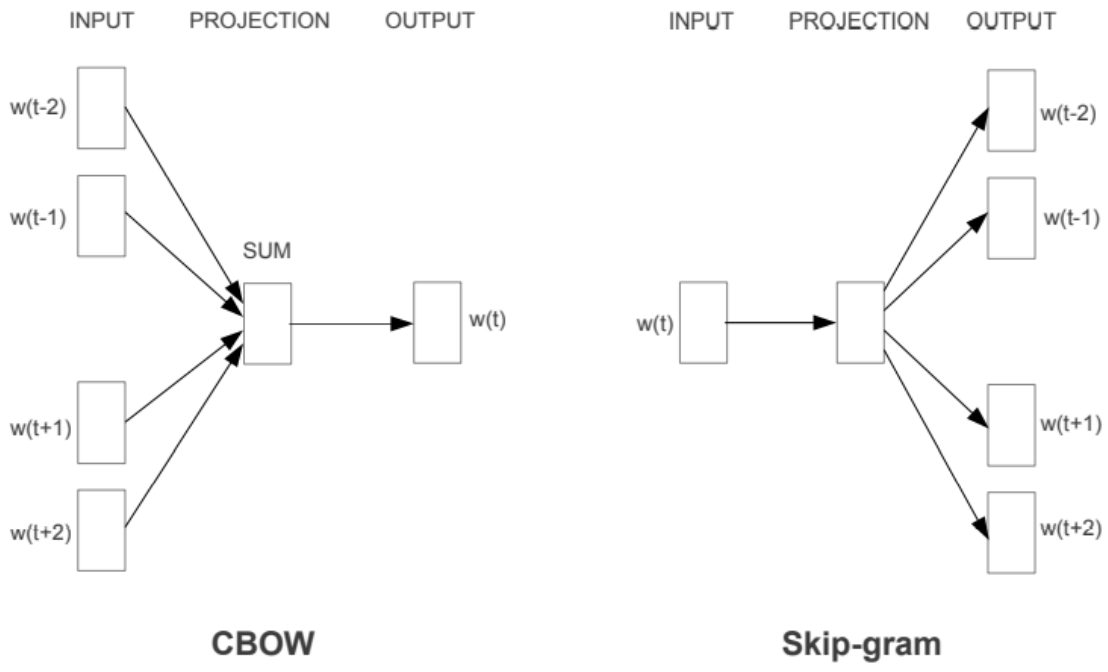

Figure 5: CBOW and Skip-gram high level diagram

3.4.4.1 Model variant used and selection criteria

The Word2Vec model was selected for sentence similarity task as its state-of-the-art word embeddings model that has proven record in generating accurate and informative vector representation of words. It can generate vectors that store both syntactic and semantic patterns within the text and so provide valuable information about sentence structure as well as meaning. The word embeddings therefore, are extremely useful for variety of NLP tasks such as sentiment analysis, sentence similarity, etc. and perform better than other such models such as BOW, TF-IDF. The other advantage is that the model can be trained on large corpus using simple and efficient genism library, or, pre-trained model can be used to generate word embeddings using the same library. The cosine similarity is considered as it's a widely used algorithm for estimating similarity between two vectors and utilized simple dot product between the two vectors.

CHAPTER 4: DATA ANALYSIS AND MODEL IMPLEMENTATION

4.1 Introduction

This section consists of details related to dataset considered for this project, the data pre-processing and data transformation techniques used, and the implementation of models used as part of proposed architecture.

4.2 Data description

As mentioned previously, dataset (Allen Institute For AI et al., 2020) consists of pdf and xml documents in jason format. The scope of this research will be restricted on the pdf documents in jason format.
Following is the breakup of folders and documents within them:

- biorxiv_medrxiv:
  - PDF - 1342 full text
- comm_use_subset:
  - PDF - 9365 full text
  - PMC - 8995 full text
- custom_license:
  - PDF - 23152 full text
  - PMC - 4773 full text
- noncomm_use_subset:
  - PDF - 2377 full text
  - PMC - 2093 full text

After looking into the dataset in python notebook, following observations have been made:

- Each file has data within seven different dictionaries. The dictionaries are as follows:
  - paper_id: Paper or file id
  - metadata: Consists of file details such as ‘Title’ of the file
  - abstract: Abstract that summarises the research content in the file
  - body_text: Research content within the respective dictionary
  - bib_entries: Bibliography
  - ref_entries: References
  - back_matter
- The dataset from Kaggle also consists of a ‘metadata’ file which consists of number of columns for each file in the dataset. The columns that will be used are as follows:
  - ‘sha’: File id
  - ‘title’: Title of the file
  - ‘abstract’: Abstract that summarises the research content in the file
- Apart from the above mentioned data, a file with 23 questions based on the three considered tasks mentioned in the Kaggle challenge (Allen Institute For AI et al., 2020) and their respective answers extracted from medical institute

sites or research papers on COVID 19 for the complete dataset considered for this research project.

- The plot for total number of records in the abstract input dataframe is as follows:

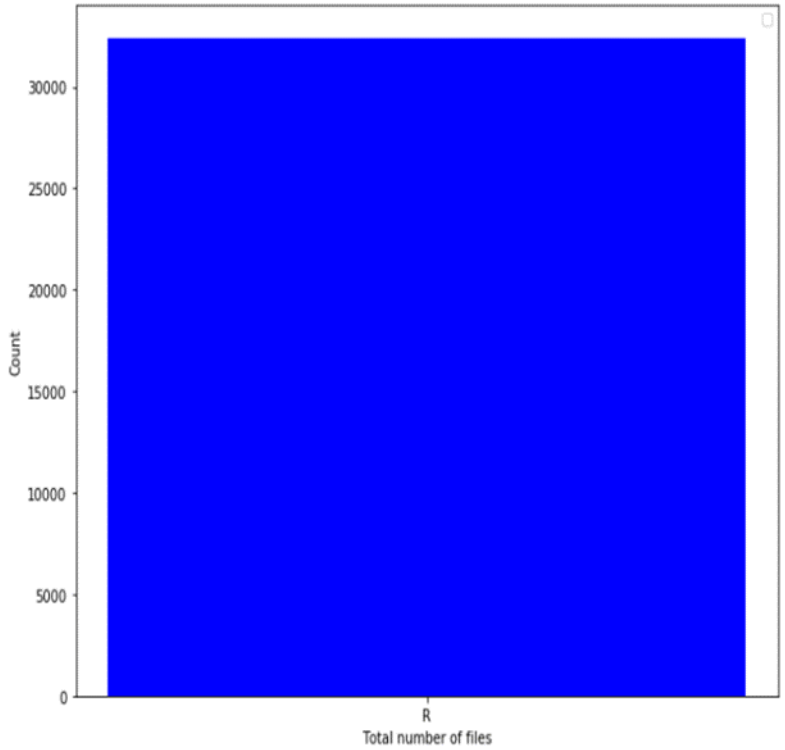

Figure 6: Total number of abstracts

- A frequency distribution plot for words in all the abstracts combined based dataset was also plotted and following is the frequency plot highlighting the top 15 words used the most in the dataset consisting of all the abstracts:

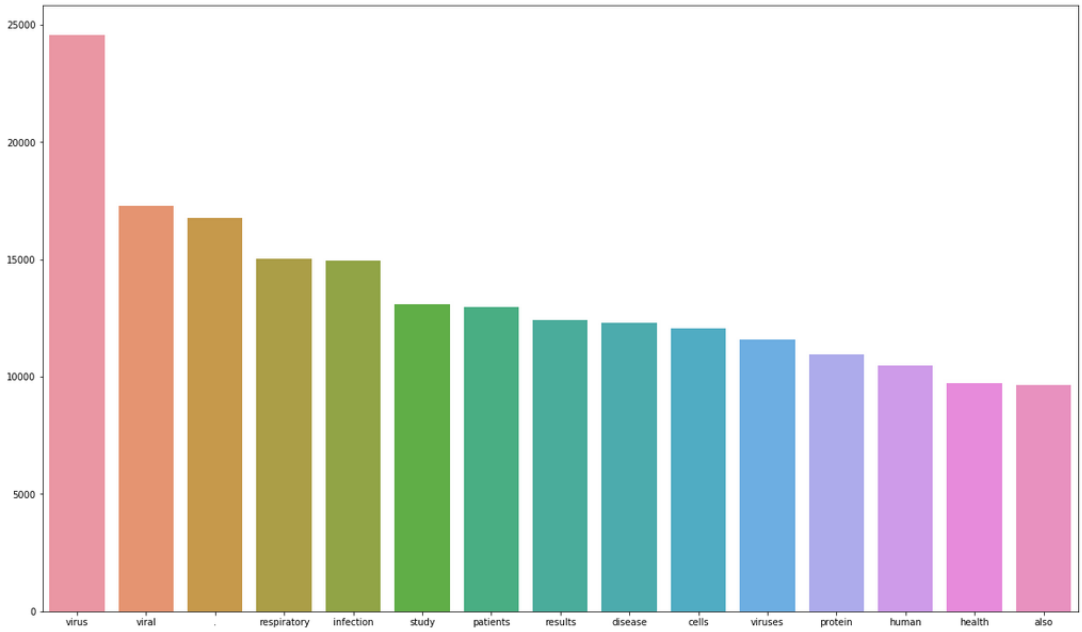

Figure 7: Plot highlighting top 15 most used words in the considered abstracts dataset

- As per the plot shown above, the word 'virus' was most common used word while other more commonly used words included words like 'viral', 'respiratory', 'infection', 'disease', etc

4.3 Data analysis and model implementations

4.3.1 Data extraction and input dataframe setup

- Relevant research content in the json pdf files is within the dictionaries “metadata”, “paper id”, “body_text” and “abstract” so information from these will be context for the document selection and extraction of respective answers with respect to task query.
- The answers will be based on “abstract” while the “body_text” from the different dictionaries within each file is consolidated and considered in the dataframe for reference for the researchers that need complete details mentioned in the documents.
- The columns that are being considered as part of the dataframe based on data extracted from files from the four folders are as per following table:

Column	**Description**
Sha	File id extracted from “paper id” dictionary within respective files
Body_text	Consolidated research content within the different ‘body_text’ dictionaries of the respective files
fl_title	Title of the respective file extracted from “metadata” dictionary within respective files
fl_abstract	The ‘abstract’ mentioned within the ‘abstract’ dictionary of the respective file

- The above mentioned dataframe is created by use a function “input_df(path)” that reads the json based pdf files from respective folders and extracts the above-mentioned columns from the files into the respective dataframes. The ‘path’ argument of the function states the path of the respective folder and the function is called four times to extract the data into respective dataframes.
- The four dataframes are then concatenated together to form single dataframe.
- The dataset from Kaggle also consists of a ‘metadata’ file which consists of number of columns for each file in the dataset. The columns that will be used are as follows:
    - ‘sha’: File id
    - ‘title’: Title of the file
    - ‘abstract’: The abstract that summarises the research content in the file
- The dataframe created from ‘metadata’ file consists of ‘34283’ file details while the one created by extracting details from files consists of ‘36236’ file details. I decided to consider the details from file records extracted from ‘metadata’ file and merged the ‘metadata’ based dataframe with the dataframe created from data extracted from files by using ‘inner join’ on column ‘sha’
- The resulting dataframe consists of columns as mentioned in the following table:

Column	**Description**
Sha	File id extracted from “paper id” dictionary within respective files
Title	Title of the respective files as mentioned in the 'metadata' file
Abstract	The ‘abstract’ mentioned within the 'metadata' file against the respective file records
Body_text	Consolidated research content within the different ‘body_text’ dictionairies of the respective files
fl_title	Title of the respective file extracted from “metadata” dictionary within respective files
fl_abstract	The ‘abstract’ mentioned within the ‘abstract’ dictionary of the respective file

- The reason the ‘title’ and ‘abstract’ values were considered from both the ‘metadata’ file as well as the dataframe with extracted values. This is because there are few entries in ‘metadata’ wherein the ‘title’ and ‘abstract’ are missing and so the idea was to populate the missing values from the respective columns ‘fl_title’ and ‘fl_abstract’ extracted from the files.
- Even after completing the above-mentioned exercise, there were few ‘null’ values in the ‘abstract’ column. I copied the ‘title’ values in the ‘abstract’ column for such instances as ‘titles’ have proven to be good replacement for missing ‘abstract’ values.
- There were few entries with missing ‘title’ values, but such entries did have ‘abstract’ values so this was not an issue as the document selection as well as answer span generation is based on ‘abstract’ value only.
- There were couple of entries with missing ‘title’ and ‘abstract’ values and these entries were deleted as I had sizeable number of records even after deletion.

- The resulting dataframe columns considered after above data selection and missing values treatment was as per the following table and consisted of 32423 records:

Column	Description
Sha	File id extracted from “paper id” dictionary within respective files
Title	Title of the respective files as mentioned in the 'metadata' file
Abstract	The ‘abstract’ mentioned within the 'metadata' file against the respective file records

- A dataframe was also created by reading the question-reference answers file “Q_A_Sheet.csv”. The file consists of tasks, questions mentioned in the Kaggle challenge (Allen Institute For AI et al., 2020) and respective reference answers collected from cited websites.
- The plot for total number of questions in the dataset is as follows:

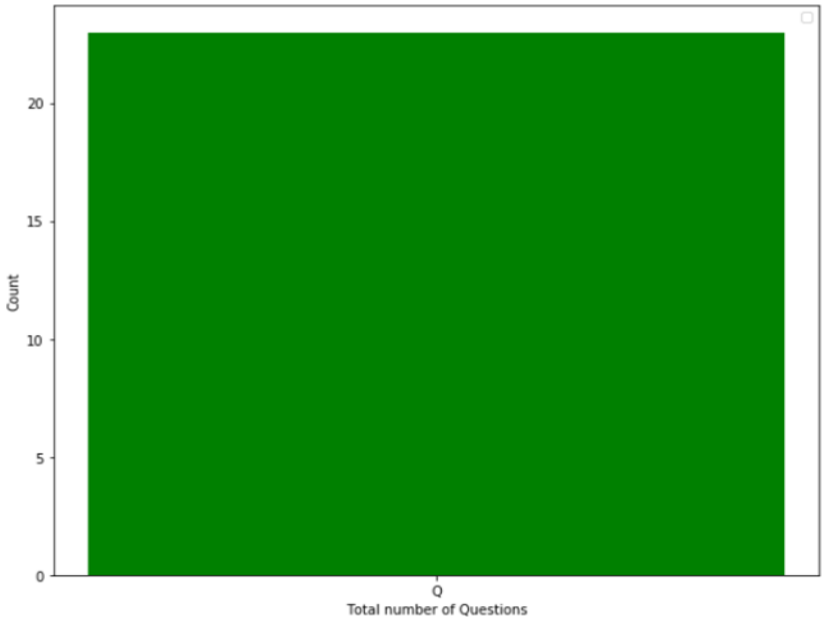

Figure 8: Total number of Questions

- The frequency distribution plot for the words used in all the questions (except stop words and irrelevant characters such as parenthesis) is shown below:

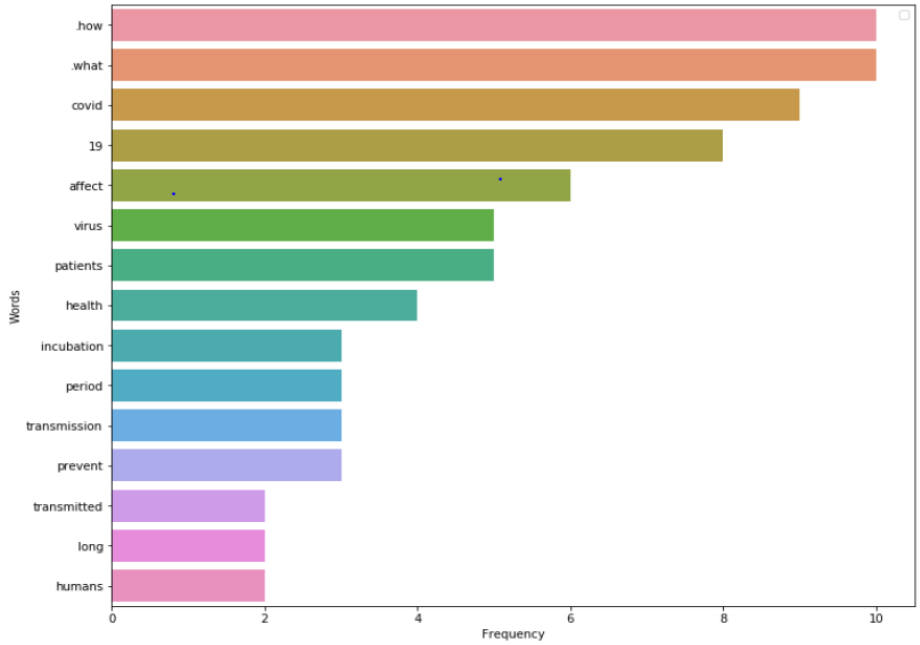

Figure 9: Plot highlighting top 15 most used words in the Questions within dataset

- As shown in the above plot, some of the most common used words in the questions (except stop words and irrelevant characters such as parenthesis) include 'how', 'what', 'covid 19', 'affect', 'virus', 'patients' and 'health'.
- The columns of the above dataframe are as per the following table:

Column	**Description**
Task	The task mentioned in the Kaggle competition
Questions	The questions with respective to the task considered
Reference Answers	The reference answers with respect to the questions collected from cited websites

4.3.2 Data processing for TF-IDF document ranking model

- Both the questions and the abstracts in the respective dataframes are processed using the user defined function “clean_document”.
- The function consisted of following widely used text processing functions:
    - **re.sub()**: This function was used to remove data such as “IP addresses”, special characters from the questions and abstracts as they do not provide useful information to the TF-IDF ranking model to aid the model in estimating relevance of a given question to the abstracts available in the respective dataframes.
    - **tokenize()**: This function converts the sentences within the abstract into individual words/tokens. The package used for the tokenize function is “TreebankWordTokenizer”.
    - **Stemmer.stem()**: This function is used for normalization of words/tokens in the abstract sentences. It reduces inflection in words to their root forms such as mapping a group of words to the same stem

even if the stem itself is not a valid word in the language. The NLTK package used is “PorterStemmer”.

- The “clean_document” function is executed for both, the questions and the abstracts, and the resulting processed texts are stored in separate columns within the respective dataframes.
- The resulting dataframe with processed abstract is as follows:

Column	Description
Sha	File id extracted from “paper id” dictionary within respective files
Title	Title of the respective files as mentioned in the 'metadata' file
Abstract	The ‘abstract’ mentioned within the 'metadata' file against the respective file records
prp_abstract	The processed abstract after execution of "clean_document" function

- The resulting dataframe with processed question is as follows:

Column	Description
Task	The task mentioned in the Kaggle competition
Questions	The questions with respective to the task considered
Reference Answers	The reference answers with respect to the questions collected from cited websites
prp_qa	The processed question after execution of "clean_document" function

4.3.3 TF-IDF document ranking model implementation

- The function “**TfidfVectorizer()**” is used to generate TF-IDF vectors for both the questions and abstract.
- The TF-IDF document ranking model essentially comprises of two user defined functions- “**td_idf_cos**” and “**doc_ranker**”.
- The function, “td_idf_cos”, calls the function “doc_ranker” which returns the abstract dataframe updated with cosine similarity scores, which are stored in column “TD_IDF_sim_score”. The updated dataframe is then sorted in descending order and a dataframe with top 10 records is then created and returned by this function
- The function “doc_ranker” calculates the similarity score between the processed query and abstract by taking cosine product between respective TD-IDF vectors. The cosine product is calculated using function “cosine_similarity” and the resulting score is the measure of document relevance with respect to the given question.
- The closer the value of similarity score is to ‘1’, the more relevant is the document with respect to given question.

4.3.4 Data transformation for BERT Question-Answer span model and implementation

- Data transformation in the case of BERT model includes following steps:
    - A code that code chunks the "abstract" into multiple smaller word segments with word overlaps on either end, so that model can better understand and check longer abstracts. This strategy was inspired from the author's proposed model in (Dirk, 2020).
    - The BERT model has its own tokenization function that was used to encode and tokenize the context and the question.
    - The BERT's tokenization function tokenizes the question and context pair together and utilizes its own '**Wordpiece**' embeddings to encode the pair.
    - The context-question encoded pair also consists of two special tokens, [CLS] and [SEP], that the BERT model utilizes for span generation and separator between the context and question.
    - The answer span tokens generated from the BERT model are then treated for removal of irregularities such as extra whitespaces (considering appropriate spacing between numerics or alphabets and ',') and presence of special characters like '##' using user defined function "cleanText".
- Two main user defined functions are utilized to get answer spans for respective questions, "**BERTSQUADPred_v2**" and "**answergen_v2**". The version-2 of both the functions are considered as they consist of updates that give better results than the first version implementations.
- The "BERTSQUADPred_v2" function consist of the abstract chunking code, tokenization of abstract-question pair and encoding the same using BERT's built in tokenization and encoding function. The "InputIds" thus generated for respective question-abstract chunk pairs are then processed by the pre-trained BERT model instance to generate the answer span start and end scores. The answer start and end points are generated using the respective scores (considering max values) and answer span is generated using the function "**CleanText**" that returns answer span after removing irregularities as mentioned above. The best answer span, out of the different possible question-abstracts chunked pairs, is then selected based on the highest 'confidence' score achieved.

  The function returns a dictionary 'answ' that consists of the 'BERT answer' which satisfies the set conditions, the original 'abstract' for which answer span is generated by BERT model and the respective 'confidence' score.
- The function 'answergen_v2' is used to generate the two output dataframes, "**df_ans_v2**" and "**qa_dt_v2**". The two dataframes are generated using the functions "td_idf_cos" and "BERTSQUADPred_v2" that are called in the outer and inner loops respectively. The function "td_idf_cos" is called in the outer loop and it generates the input dataframe that consists of 10 records of

the most relevant abstracts with respect to a given question. This loop is executed till all the questions are exhausted. The “BERTSQUADPred_v2” function is called in the inner loop that returns the above-mentioned dictionary for the each of the 10 abstracts with respect to the question considered in the loop.

The dictionaries with positive ‘confidence’ score, ‘BERT answer spans’ not starting with preposition or connecting words (‘will’, ‘on’, ‘of’, etc) and non-null ‘BERT answer spans ’ are considered and the output dataframe, “df_ans_v2”, is populated with the considered question in the loop, the filtered answer spans, respective abstracts and respective “confidence scores” for the all the questions in the dataset.

- The output dataframe “df_ans_v2” has the following structure:

Column	Description
brt_sha	File id extracted from “paper id” dictionary within respective files
brt_ans	The answer span generated by BERT model for a given abstract and question
brt_conf	The "confidence" score achieved by the model while generating the answer span
brt_abst	The abstract considered for answer span generation
brt_question	The question considered for answer span generation

4.3.5 BERTSUM summarizing model implementation

- The function ‘answergen_v2’ also consists of code to join the multiple answer spans generated for a given question and shortlisted abstracts. The resulting list consisting of all the possible answer spans is then passed as input to the pre-trained BERTSUM model instance to get the summarized answer.
- The reference answers are also summarised using similar conditions and model configurations as used for the summarization of combined abstract answers.
- The summarized answers for respective questions are stored in the second output dataframe, “qa_dt_v2”. The structure of dataframe, “qa_dt_v2”, is as follows:

Column	Description
Task	The task mentioned in the Kaggle competition
Questions	The question with respect to the task considered
Reference Answers	The reference answer collected from the cited websites with respect to the question
prp_qa	The processed question after execution of "clean_document" function
Model_Sum_Answer	The summarized answer from the BERTSUM model
Sum_Reference_answers	Reference answers summarized by BERTSUM model

4.3.6 Word2Vec word embeddings model implementation

- The Word2Vec embeddings model from gensim library is used and then its trained with the words within all the abstracts in the dataset. This helps the model instance to build vocabulary based on dataset being used for this project.
- The trained model instance is then used to create embeddings or vectors for the summarized joined answer and respective summarized reference answer.
- The cosine product of the two will then be calculated to measure the similarity.
- A user defined function, "wrd2vec_cos()", is used to implement the above two steps for all the model generated summary answer-summarized reference answer pairs.
- The output dataframe, "qa_dt_v2", is updated with the respective similarity scores and the structure of resulting output dataframe is as follows:

Column	**Description**
Task	The task mentioned in the Kaggle competition
Questions	The question with respect to the task considered
Reference Answers	The reference answer collected from the cited websites with respect to the question
prp_qa	The processed question after execution of "clean_document" function
Model_Sum_Answer	The summarized answer from the BERTSUM model
Sum_Reference_answers	Reference answers summarized by BERTSUM model
similarity_score	Similarity score calculated using Word2Vec model and cosine similarity algorithm

4.4 Summary

The dataset considered was described in detail and the relevant data extraction, pre-processing as well as transformation methods like 'tokenization', 'stemmer', 'stopwords' removal and BERT tokenization and encoding were detailed. The implementation of the three models – TF-IDF based document ranker, pretrained BERT model and the BERTSUM summarizing model was also detailed. The TF-IDF document ranker model generated the input dataframe consisting of top 10 abstracts out of the available abstracts in the dataset, ranked based on cosine similarity with respect to the question. The pretrained BERT model generated answer spans considering the 10 abstracts for the question being considered. The filtered answer spans are then stored in one of the output dataframes, along with other mentioned details. The pretrained BERTSUM model instance is then used to summarize multiple answer spans and the summarized answers for the respective questions are stored in the second output dataframe, along with other mentioned details. The similarity scores for respective 'joined' abstract answer spans summary and reference answer summary pairs are then calculated using the Word2Vec model and cosine similarity algorithm.

CHAPTER 5: RESULTS AND SUMMARY

5.1 Introduction

In this section the results achieved using models and algorithms utilized within the model architecture will be detailed. The results will be mentioned within the respective model specific sub sections.

5.2 TD-IDF document ranking model results

- The similarity scores that were generated using the TF-IDF vectorizer model and cosine similarity function helped in shortlisting top 10 most relevant documents along with their respective abstracts for most questions.
- This was evident with the fact that the BERT model used for answer span generation had relevant abstracts to extract answer spans in most cases.
- There was no specific 'ideal' score identified for TD-IDF as it was observed that the TD-IDF similarity scores varied substantially for different abstract and question pairs. Also, it was not necessary that higher TD-IDF similarity score for an abstract resulted in better answer spans being generated from it.
- This could be due to abstracts with more relevant data being larger in size and hence, consisting of more redundant information with respect to a given question as compared to the smaller abstract with higher score.
- The cases where the TF-IDF vectorizer-cosine similarity model failed to shortlist relevant abstracts were when **titles** were used instead of **abstracts** due to missing abstracts. The possible reason for this is that the titles do not necessarily consist of relevant information required to answer questions based on particular object or piece of research covered in detail with the document or its abstract. Therefore, titles do not help in generating valuable vector information with the use of TD-IDF that could be matches with respective question vectors.
- The model also struggled to shortlist relevant documents for question such as **"Can animals transmit COVID 19?"**. This most likely could be due relevant documents having the missing abstracts as well as the abstracts not necessarily consisting of details related to 'animal' infection. The word 'animal' though did occur in the abstracts as observed during data analysis.
- The output dataframe, "**df_ans_v2**", consists of details such as answer spans, the respective abstracts and questions and the records within it show that the abstracts shortlisted were mostly relevant with respect to the questions.
- Following are few examples of records in one of the output dataframes, "df_ans_v2" that was extracted in the form of "**df_ans_v2.xlsx**" file is embedded in this report as one of the appendices.

B	C	D	E	F
brt_sha	brt_ans	brt_conf	brt_abst	brt_question
f26952fcfc	the transmission modes of sars-coronavirus appear to be through droplet spread, close contact and fomites	5.0821805	the transmission modes of sars-coronavirus appear to be through droplet spread, close contact and fomites although air borne transmission has not been ruled out. this clearly places dental personnel at risks as they work in close proximity to their patients employing droplet and aerosol generating procedures. although the principle of universal precautions is widely advocated and followed throughout the dental community, additional precautionary measures – termed standard precaution may be necessary to help control the spread of this highly contagious disease. patient assessment should include questions on recent travel to sars infected areas and, contacts of patients, fever and symptoms of respiratory infections. special management protocols and modified measures that regulate droplet and aerosol contamination in a dental setting have to be introduced and may include the reduction or avoidance of droplet / aerosol generation, the disinfection of the treatment field, application of rubber dam, pre-procedural antiseptic mouthrinse and the dilution and efficient removal of contaminated ambient air. the gag, cough or vomiting reflexes that lead to the generation of aerosols should also be prevented.	Is the COVID virus transmitted by aerosol, droplets, food, close contact, faecal matter, or water?

Figure 10: Example of abstract shortlisted consisting of relevant content wrt question

d87523b85	patients with respiratory symptoms are at greater risk of covid-19 transmission	4.097403397	patients with respiratory symptoms are at greater risk of covid-19 transmission	How does weather, heat, and humidity affect the transmission of COVID 19?

Figure 11: Example of abstract shortlisted not relevant to the question, abstract being the title of document

5.3 BERT and BERTSUM model results

- The two pretrained models, BERT and BERTSUM, were implemented using two versions of user defined functions "**BERTSQUADPred**" and "**answergen**".
- The first versions of the two functions had '**overlap**' factor as 1.1, limited answer span filtering conditions and static BERTSUM parameters such as ratio of sentences to be considered for summary was kept as **0.4** irrespective of length of the combined answer spans.
- The observations made after analysing the two output dataframes, "dt_ans" and "qa_dt", generated by executing the above two mentioned functions are as follows:
    - The proposed model architecture seemed to have generated relevant answers for most questions.
    - However, there are few questions such as "**Can the virus be transmitted asymptomatically or during the incubation period**?"

where the proposed BERT model was not able to generate meaningful spans from the shortlisted abstracts, while, there was no span generated for few queries such as "**What do we know about virus genetics, origin, and evolution?**".

- In the case of few queries the generated answers start with incomplete sentences or parenthesis, highlighting model not being able to understand the context for the given query and generate proper span.
- The relevant answer spans generated were semantically aligned with respective questions but were not always syntactically perfect.
- The performance of pretrained BERT model after execution of first version of the functions was not perfect but it did generate meaningful spans.
- The confidence scores achieved by BERT model for the answer spans did not always give accurate estimation of model's performance as they were higher for shorter answers that did not necessarily consist of complete information required to answer the question. Therefore, they could not be considered as reliable evaluation metric.
- The BERTSUM summarization model generated the summaries based on the abstract spans that were joined and passed to the BERTSUM model. The summaries were not always perfect, and the lengths were not consistent as the fixed ratio of sentences to be considered meant that few summaries lost important information.

- The second version of the two user defined functions were updated with code to filter answer spans generated by BERT model that started with preposition or connecting words ('will', 'at', 'and', 'from', etc), parenthesis such as -')', ' (' and had length less than 20. The **overlap** factor was kept as **1.2** in this iteration and BERTSUM summary model code was also updated with dynamic values of ratio of sentences to be considered set with respect to size of the input to BERTSUM model and the minimum length of sentence to be considered was set as '30'. The code was also added to summarize the reference answers as well. Following are the observations that were made after analysing the two output dataframes generated in the second iteration:
  - The proposed model architecture seemed to have generated relevant answers for most questions again and with updates, removed parenthesis such as ')' from start of the answer.
  - However, for the same questions such as "**Can the virus be transmitted asymptomatically or during the incubation period?**" the proposed BERT model with updates to functions was still not able to generate meaningful spans from the shortlisted abstracts, while, there was no span generated for the same queries such as "**What do we know about virus genetics, origin, and evolution?**".
  - The results from second iteration highlighted that updating the '**overlap**' factor did not give meaningful improvements as model was

still not able to understand the context for the given query and generate proper span for the same queries as in the first iteration. Some cases of answer spans not being generated can be attributed to the shortlisted documents not consisting of relevant information, mostly due to **missing abstracts** and use of respective **titles** instead. However, there are cases when relevant abstracts were selected but the model failed to span specific details such as impact of **'asymptomatic patients'** on the COVID-19 spread, highlighting the limitations of using untuned pre-trained model over a dataset focussed on medical research on a recent pandemic such as COVID-19.

    - This means that to improve the performance, proposed BERT model will need to be fine-tuned with help of larger queries/reference answers data base along with the existing large context documents database. This is however, not in scope of this research due to paucity of time required to collect data and fine tune the model
    - The filtering condition of removing answer spans starting with prepositions or connecting words such as "on", "at", "from", "will", etc helped getting better answer spans from the model
    - The changes to the way summarization of joined answers was carried by the model helped in improving the quality of summarized answers. The setting of minimum length for sentence to be considered helped in only valid sentences to be considered, while, the dynamic 'sentence ratio' settings based on the length of the joined answer made sure relevant information was retained.
    - The summarization of "reference answers", using similar conditions as considered for the joined answers, helped in resizing them to reasonable lengths when compared to respective summarized answers without losing relevant information.

- The plot showcasing the total number of model generated span-based summaries and the total number of questions is as follows:

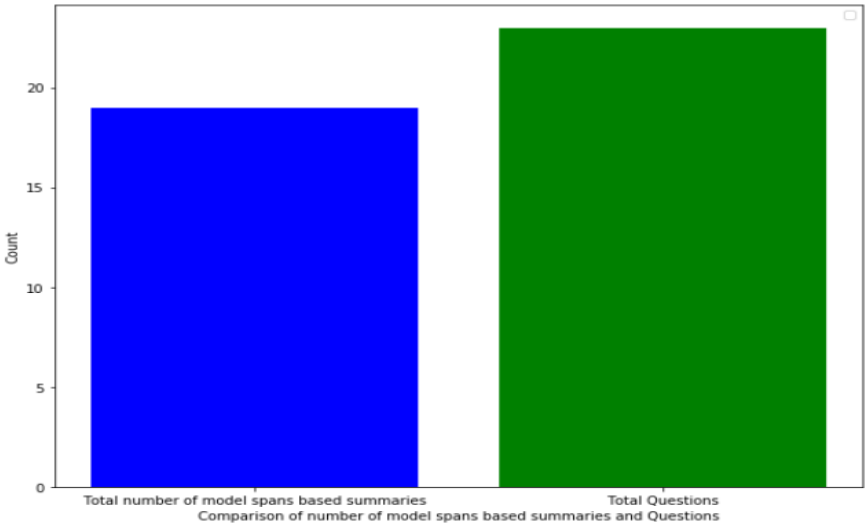

Figure 12: Comparison of number of model generated span-based summaries and Questions

- The plot to show top 15 most words mentioned in the answer span-based summaries along with their respective frequencies is shown below:

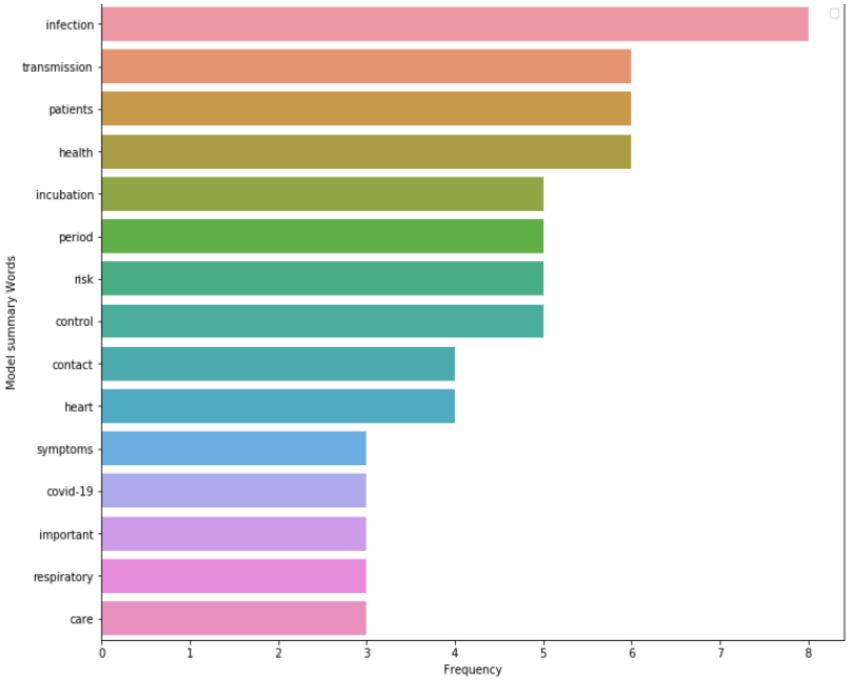

Figure 13: Plot highlighting top 15 most used words in the answer span-based summaries

- The above plot shows that some of the most mentioned words in the model answer span-based summaries include 'infection', 'transmission', 'patients', 'health', 'incubation', 'period', 'risk', 'heart', 'covid-19'.
- The plot to show top 15 most words mentioned in the reference answer-based summaries along with their respective frequencies is shown below:

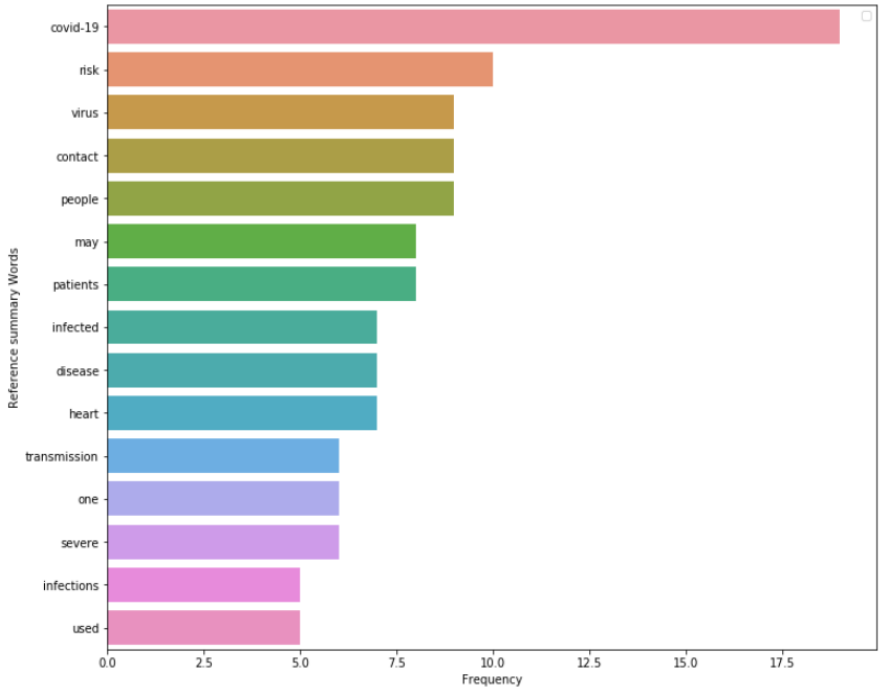

Figure 14: Plot highlighting top 15 most used words in the reference answer-based summaries

- The above plot shows that some of the most mentioned words in the reference answer-based summaries include 'covid-19', 'risk', 'virus', 'infected', 'transmission', 'patients', 'heart', 'contact', 'people'.
- It can be seen that both the answer span-based summaries and the reference answer summaries share the most common words used, which shows that it's safe to conclude that both the answers are more often than not are semantically aligned.
- The complete results of the two dataframes can be read via the two sheets, "**df_ans_v2.xlsx**' and "**qa_dt_v2.xlsx**", attached as appendices.

5.4 Word2Vec based sentence similarity model results

- The Word2Vec embeddings and cosine similarity-based model was executed in two variants. One without Smooth Inverse function (SIF) and the other with SIF.
- The results observed after first iteration are as follows:
    - The cosine similarity scores achieved were high except for questions that had '0' summarized answers and in one case wherein the summarized answer was short and did not answer the question ("What is the fatality rate of COVID 19?") completely.
    - High similarity scores were achieved even in the cases wherein the summarized answers were not syntactically perfect but did consist of enough information related to the respective questions. This showed that semantic meaning of the summarized answers with respect to the questions was given importance.

    - It was observed that high similarity score was achieved in a case wherein a random sentence was considered along with a summarized reference answer that prompted me to try and improve performance of this model by using SIF
- The results observed after second iteration are as follows:
    - It was observed that using SIF had made Word2vec model lot more aggressive in checking semantic similarity between the two answers considered for evaluation.
    - The difference in content length, quality in terms of grammatical correctness of sentences was considered to an extreme, resulting in score that was not necessarily depicting the true variance between the two sentences.
    - The similarity score for the case wherein one content is a random sentence and the other is one of the reference answers was higher than when the model generated summary answer was compared to the reference answer to the same question, even though the content in both of them was quite similar. This showed that SIF is not ideal for the dataset being considered for this project.
    - Therefore, the original implementation of Word2Vec and cosine similarity sentence similarity check model was considered.
- The following plot shows the number of cases wherein the similarity scores are above 0.95 or below 0.95. It can be seen that for 18 out of the 19 questions that the proposed model did generate summaries, the score achieved in 0.95.

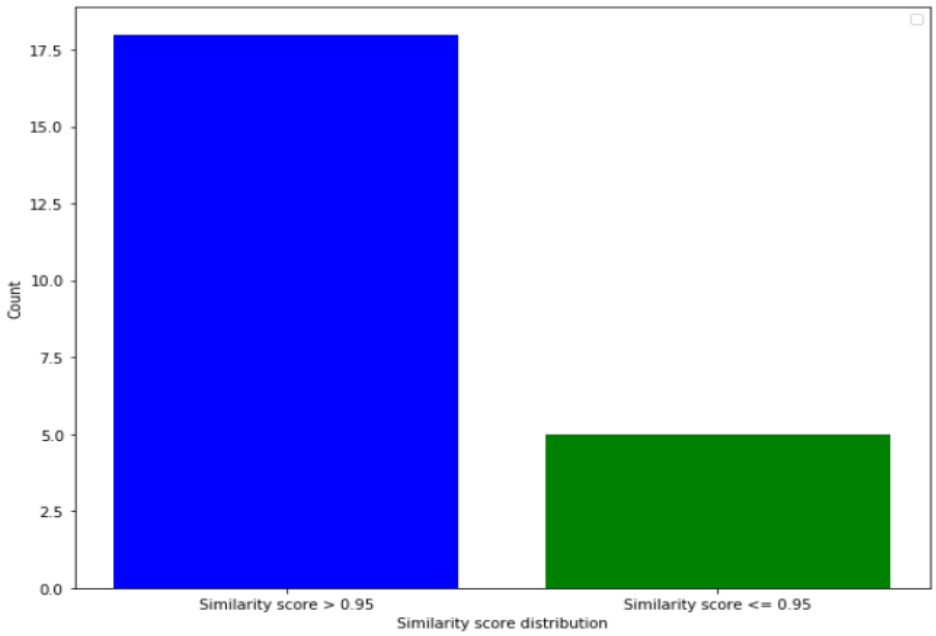

Figure 15: Similarity score distribution

- The plot below shows the sizes of the answer span based summaries, reference answer summaries and respective similarity scores.

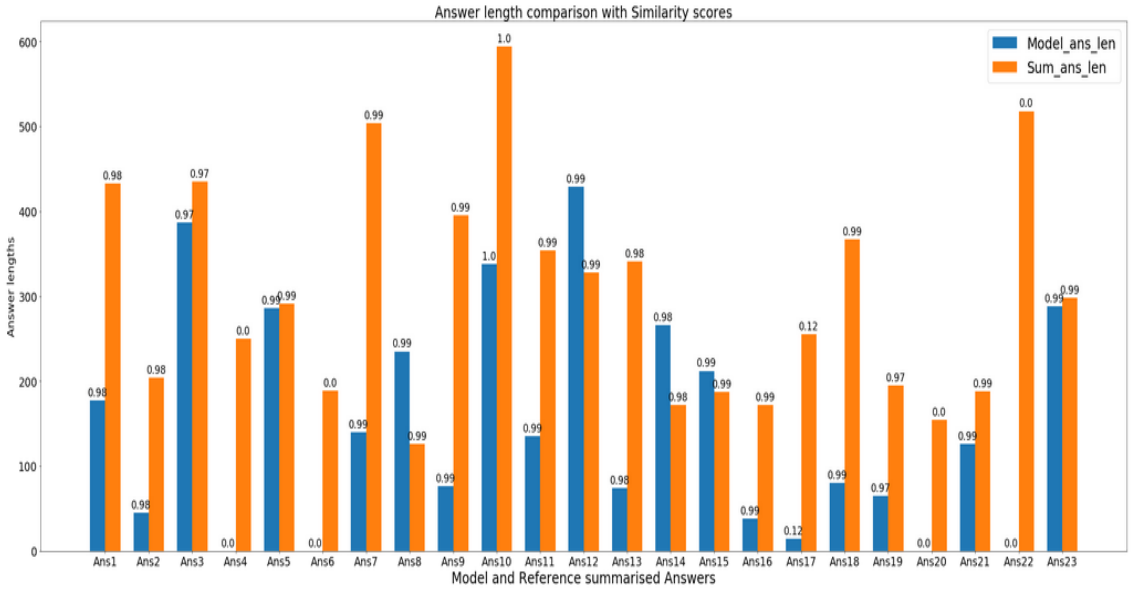

Figure 16: Answer length comparison with Similarity scores

- The plot shows that size difference between the two answers does not necessarily impact the similarity scores till both answers are big enough to capture important information, making them similar symantically.
- The similarity scores are '0' for all the cases where answer span based summaries are not generated and the score is low for an instance (Ans17) when the answer span based summary is too small and does not have enough information to match the reference answer summary.
- The **cut off similarity score** for considering the proposed model generated summaries being semantically similar to the respective summarized reference answers is considered to be **0.97**, based on high scores achieved within the considered dataset.
- This would also be evaluation metric considered to determine the quality of summaries generated by proposed model architecture.
- The results of the Word2Vec-cosine sentence similarity model can be reviewed in the form of similarity scores mentioned in the attached "**qa_dt_v2.xlsx**" appendix.

5.5 Summary

The results achieved using the selected models and algorithms within the proposed hierarchical model architecture were detailed. The observations based on the various iterations of the models were mentioned along with respective inferences. The proposed hierarchical model gave reasonable performance in generating summaries for most questions but there were few questions for which summaries were not generated. It was observed that using pre-trained BERT model for answer span generation has limits as it missed generating meaningful spans for few questions even though the shortlisted abstracts did have content relevant to the questions. The use of

fine-tuned model could also give more syntactically correct answers as well. The TD-IDF based document shortlisting model performed well for most questions but struggled when titles were used in the case of missing abstracts within the dataset. The pre-trained BERTSUM model generated better summaries while using dynamic ratio of considered sentences parameter and the minimum length of sentence set as ‘30’. The sentence similarity scores achieved by Word2Vec-cosine similarity-based model was high for most cases, apart from cases with no generated summaries and case when generated summary did not have complete information. The similarity scores achieved are considered as evaluation metric for proposed model architecture as well and mostly high scores show reasonable model performance, considering the use of only the pre-trained models.

Chapter 6: CONCLUSION AND FUTURE RECOMMENDATIONS

6.1 Introduction

The aim for developing a hierarchy levels based model capable of answering task based questions by learning from the dataset shared in the Kaggle challenge (Allen Institute For AI et al., 2020) was achieved. In this chapter the results of the proposed model architecture are discussed, and future scope of work is detailed.

6.2 Discussion and conclusion

The aim of this research was primarily to propose a hierarchy layers based model capable of answering task based questions by learning from the dataset shared in the Kaggle challenge (Allen Institute For AI et al., 2020). This challenge was selected as COVID19 has disrupted lives and operations across the globe and any help as a data scientist in extracting relevant information with respect to research questions in the need of the hour. The goal included using **TF-IDF and cosine similarity**-based document ranker model as the top 10 most relevant documents based on abstracts ranker model with respect to the questions. The state of the art transformer models such as **BERT** (Devlin et al., 2019) and **BART** (Lewis et al., 2020) were then used to generate answer spans and then summarize the spans to make sure final answer from the proposed model architecture consists of all the relevant information extracted within different spans by the pre-trained BERT model. Additional objective included collecting reference answers with respect to the task-based questions within the dataset from cited websites as the dataset considered did not have the answers. As there were limited questions and no answers within the dataset, the implementation of pre-trained BERT and BERTSUM models was considered for this project scope. The **Word2Vec-cosine similarity** model was used to calculate similarity scores for the model generated summary and reference answer summary pairs and these scores were considered as evaluation metric for the proposed model's performance.

The analysis of dataset revealed the structure of the research documents and the relevant content to be considered for answer summaries generation. The metadata file within the dataset was used as well as only the file records mentioned in the metadata file were considered. The complete research body within each file as well as the respective abstracts were also extracted, though the abstracts were considered for both document ranker model and answer span generation model. The extracted abstracts were used to populate the metadata file based dataframe wherever there were missing abstracts. The title of the file was used as the abstract in cases where abstract was missing in both the file as well as the metadata file. The few records where both title and abstract entries were missing were removed and the required input dataframe consisting of required details was achieved. The second part of input was the dataframe based on the tasks, respective questions and the reference answers.

The data (abstracts and questions) processing for TF-IDF model, as mentioned in the section 4.3.2, included removal of irrelevant characters, tokenizing the content into

words, removing the ‘stop words’ and extra spaces, stemming the words to respective root words and then returning the joined content for TF-IDF vectorization. The data processing for the BERT model included tokenization and encoding the question and the chunked abstract pair using the BERT tokenizer and encoder. This is detailed in section 4.3.4 of this report.

The use of **TF-IDF** and cosine similarity-based document ranking model proved effective as the 10 abstracts ranked by the model was relevant for most of the questions as per the observations mentioned in the section 5.2. The cases where the model failed to rank relevant abstracts were when titles were used instead of abstracts due to missing abstracts. The model also struggled to shortlist relevant documents for question such as “"Can animals transmit COVID 19?". This most likely could be due to relevant documents having the missing abstracts as well as the abstracts not necessarily consisting of details related to ‘animal’ infection. The solution to such scenarios could be use of the content in the complete research body rather than the titles as that would contain the relevant details. However, such a solution strategy was not part of this project scope.

The performance achieved for the pre-trained **BERT** model as detailed in section 5.3 was reasonable but not perfect. The answer spans generated after second iteration with updated functions gave better performance with answer spans that were semantically related to the questions and were also without some of the syntactic irregularities. However, in the same few cases answer spans were not generated at all, in some cases there were no relevant shortlisted documents but there were couple of cases where relevant abstracts were present. The limitations of using pre-trained BERT model without fine-tuning were observed in such scenarios as the model failed to understand the semantic meaning within content focused on medical issues such as COVID19.

The performance of the **BERTSUM** summarizer model was also better in the second iteration with updated functions as mentioned in the section 5.3. The model gave summaries with most relevant details within the combined abstracts and the length of summaries was consistent too. The quality of summaries though depends mostly on the answer spans generated so better the results from BERT model, better would be the quality of the summaries.

The **Word2Vec** and cosine similarity-based sentence similarity model was used to calculate the similarity scores for the pairs of model generated summary and reference answer. The model gave more reliable similarity scores without use of SIF as mentioned in the section 5.4. The scores achieved were on the higher side for most cases but was low in the case of an incomplete answer and ‘0’ for cases when model failed to generate summaries. The similarity scores are considered to be the evaluation metric for the quality of summaries generated and **high score of 0.97** is considered to be the passing score for summary to be considered relevant with respect to the question.

The two output dataframes generated as the output of the proposed model architecture were extracted as excel files “**qa_dt_v2.xlsx**” and “**df_ans_v2.xlsx**” and are attached as appendices for reference.

The research questions that were considered as part of this project were answered as follows:

- The evaluation metric considered was the similarity score generated for the pairs of model generated summary based on the answer spans retrieved from model and the reference answer summaries. A high score of 0.97 was considered as cut off score for model generated summary to be considered as relevant or similar to the reference answer summary for a given question.
- The granularity level selected was **document level** as **abstracts** for each of the document were passed to the model as an input, along with respective question. This decision was based upon the capability of model to parse data as well as the large number of documents within the dataset.
- There was no specific '**ideal**' score identified for TD-IDF as it was observed that the TD-IDF similarity scores varied substantially for different abstract and question pairs. Also, it was not necessary that higher TD-IDF similarity score for an abstract resulted in better answer spans being generated from it. Therefore, it was decided to consider the top 10 abstracts and then summarize the filtered answer spans retrieved from the BERT model.

6.3 Contribution to research field

- The proposed hierarchy layers-based question-answer model has shown to be effective tool for research scientists and analysts to extract relevant information from dataset based on huge number of research documents. This should help get the medical fraternity to understand the COVID-19 virus better, find possible cures and develop strategies to limit or stop the spread in faster, efficient and collaborative way.
- The use of TD-IDF and cosine similarity as a document ranker model layer within the proposed model architecture highlights the versatility of TD-IDF algorithm.
- The use of BERTSUM summarization to extract relevant information from pretrained, but untuned, BERT model generated answer spans instead of considering just highest confidence score based single answer span proved to be effective strategy of avoiding loss of important information and a good use case for BERTSUM based summarization. This is because the confidence scores achieved were not reliable due to BERT model not being fine-tuned.
- The optimisation code used for BERTSUM wherein the model considered ratio of relevant sentences based on the size of abstracts was a useful update to summarization model that can be used in other summarization tasks as well.
- The used of Word2Vec and cosine similarity model as evaluation metric for unsupervised question-answer model performance proved to be an effective strategy and good use case for Word2Vec model.

6.4 Future Work

- One of the tasks as part of future work is to fine-tune the BERT answer generation model with dataset consisting of more questions and respective answers. This should optimise the BERT model and help in getting better answer spans and possibly get answer spans for questions that were missed by the model in this project.
- The use of the proposed model architecture on other datasets focussed on question-answer task.
- Evaluate the performance of the proposed model architecture considering top-15 documents instead of top 10 that were considered in this project. This would increase the probability of finding correct context with respect to the questions and hence, improve the model performance.
- The use of the complete research body within the files in cases where abstract is not available for both the document ranker and answer span models. This should solve the issues faced by TD-IDF and cosine similarity-based document ranker model in ranking the relevant documents when titles were used in place of missing abstracts.
- The implementation of different models such as Anserini search engine (Yang et al., 2018) as document ranker model and compare the performance on the same dataset.
- The implementation of BART (Lewis et al., 2020) as summarization model and compare the performance.

REFERENCES

ACOG, (2020) *Coronavirus (COVID-19), Pregnancy, and Breastfeeding | ACOG*. [online] Available at: https://www.acog.org/patient-resources/faqs/pregnancy/coronavirus-pregnancy-and-breastfeeding#How does COVID19 affect pregnant women [Accessed 4 Oct. 2020].

Allen Institute For AI, Goldbloom, A., Lin, P., Mooney, P., Carissa, S., Kohlmeier, S., Devrishi, Bozsolik, T. and Hamner, B., (2020) *COVID-19 Open Research Dataset Challenge (CORD-19) | Kaggle*. [online] Available at: https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=568 [Accessed 4 Sep. 2020].

Bahdanau, D., Cho, K.H. and Bengio, Y., (2015) Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp.1–15.

Bengio, Y., (2003) A Neural Probabilistic Language Model Yoshua. *Fullerenes Nanotubes and Carbon Nanostructures*, 268, pp.465–470.

Çelikyilmaz, A., Bosselut, A., He, X. and Choi, Y., (2018) Deep Communicating Agents for Abstractive Summarization. *CoRR*, [online] abs/1803.10357. Available at: http://arxiv.org/abs/1803.10357.

Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A. and Berant, J., (2017) Coarse-to-fine question answering for long documents. In: *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. [online] pp.209–220. Available at: http://arxiv.org/abs/1611.01839.

Davies, N.G., Klepac, P., Liu, Y., Prem, K., Jit, M., Pearson, C.A.B., Quilty, B.J., Kucharski, A.J., Gibbs, H., Clifford, S., Gimma, A., van Zandvoort, K., Munday, J.D., Diamond, C., Edmunds, W.J., Houben, R.M.G.J., Hellewell, J., Russell, T.W., Abbott, S., Funk, S., Bosse, N.I., Sun, Y.F., Flasche, S., Rosello, A., Jarvis, C.I. and Eggo, R.M., (2020) Age-dependent effects in the transmission and control of COVID-19 epidemics. *Nature Medicine*, [online] 268, pp.1205–1211. Available at: https://doi.org/10.1038/s41591-020-0962-9 [Accessed 3 Oct. 2020].

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. pp.4171–4186.

Dietz, L., Verma, M., Radlinski, F. and Craswell, N., (2012) TREC Complex Answer Retrieval Overview. pp.1–13.

Dinan, E., Logacheva, V., Malykh, V., Miller, A.H., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., Prabhumoye, S., Black, A.W., Rudnicky, A.I., Williams, J., Pineau, J., Burtsev, M.S. and Weston, J., (2019) The Second Conversational Intelligence Challenge (ConvAI2). *CoRR*, [online] abs/1902.00098. Available at: http://arxiv.org/abs/1902.00098.

Dirk, (2020) *Anserini+BERT-SQuAD for Semantic Corpus Search | Kaggle*. [online] Available at: https://www.kaggle.com/dirktheeng/anserini-bert-squad-for-semantic-corpus-search/notebook [Accessed 25 Aug. 2020].

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.W., (2019) Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32NeurIPS.

Durrett, G., Berg-Kirkpatrick, T. and Klein, D., (2016) Learning-Based Single-

Document Summarization with Compression and Anaphoricity Constraints. *CoRR*, [online] abs/1603.08887. Available at: http://arxiv.org/abs/1603.08887.
Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J. and Auli, M., (2019) {ELI}5: Long Form Question Answering. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. [online] Florence, Italy: Association for Computational Linguistics, pp.3558–3567. Available at: https://www.aclweb.org/anthology/P19-1346.
Forster, P., Forster, L., Renfrew, C. and Forster, M., (2020) How genomic epidemiology is tracking the spread of COVID-19 locally and globally. *Proceedings of the National Academy of Sciences of the United States of America*, [online] 11717, pp.9241–9243. Available at: https://cen.acs.org/biological-chemistry/genomics/genomic-epidemiology-tracking-spread-COVID/98/i17 [Accessed 4 Oct. 2020].
Group, S.H.C.I., (2015) Daemo : a Self-Governed Crowdsourcing Marketplace. pp.2–3.
HANSON ER, (1971) MUSICASSETTE INTERCHANGEABILITY. THE FACTS BEHIND THE FACTS. *AES: Journal of the Audio Engineering Society*, 195, pp.417–425.
Harvard Health Publishing, (2020) *Preventing the spread of the coronavirus - Harvard Health*. *Harvard Health publishing*. Available at: https://www.health.harvard.edu/diseases-and-conditions/preventing-the-spread-of-the-coronavirus [Accessed 4 Oct. 2020].
Hermann, K.M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M. and Blunsom, P., (2015) Teaching Machines to Read and Comprehend. *CoRR*, [online] abs/1506.03340. Available at: http://arxiv.org/abs/1506.03340.
Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M. and Berthelot, D., (2015) : A Novel Large-scale Language Understanding Task over Wikipedia.
Huang, Y., Sun, M. and Sui, Y., (2020) How Digital Contact Tracing Slowed Covid-19 in East Asia. *Harvard Business Review Digital Article*, [online] pp.1–8. Available at: https://hbr.org/2020/04/how-digital-contact-tracing-slowed-covid-19-in-east-asia [Accessed 5 Oct. 2020].
Initiative, A.I. for A. in partnership with the C.Z., (2020) COVID-19 Open Research Dataset Challenge (CORD-19). *Semantic Scholar*. [online] Available at: https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=568 [Accessed 6 Jul. 2020].
Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L., (2020) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. pp.7871–7880.
Li, W., Li, W. and Wu, Y., (2018) A unified model for document-based question answering based on human-like reading strategy. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. [online] pp.604–611. Available at: www.aaai.org.
Liu, Y., (2019) Fine-tune BERT for Extractive Summarization. [online] Available at: http://arxiv.org/abs/1903.10318.
Liu, Y. and Lapata, M., (2020) Text summarization with pretrained encoders. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp.3730–3740.
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. [online] 1. Available at: http://arxiv.org/abs/1907.11692.

Loye, G., (2019) Attention mechanism. *FLOYDHUB*, [online] pp.0–16. Available at: https://blog.floydhub.com/attention-mechanism/ [Accessed 6 Jul. 2020].
Luong, M.T., Socher, R. and Manning, C.D., (2013) Better word representations with recursive neural networks for morphology. *CoNLL 2013 - 17th Conference on Computational Natural Language Learning, Proceedings*, pp.104–113.
Mikolov, T., (n.d.) Language Models for Automatic Speech. 4.
Mikolov, T., Chen, K., Corrado, G. and Dean, J., (2013) Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp.1–12.
Mikolov, T., Deoras, A., Povey, D., Burget, L. and Černocký, J., (2011) Strategies for training large scale neural network language models. *2011 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2011, Proceedings*, pp.196–201.
Mikolov, T., Jiri, K., Burget, L., Glembek, O. and Cernocky, J. 'Honza', (n.d.) NEURAL NETWORK BASED LANGUAGE MODELS FOR HIGHLY INFLECTIVE LANGUAGES ˇ Tom ´ a s ˇ Mikolov , Jiˇ r ´ ı Kopeck ´ y , Luk ´ a s ˇ Burget , Ondˇ rej Glembek and Jan " Honza " Cernock ´ y Speech @ FIT , Faculty of Information Technology , Brno University of T. pp.2–5.
Narayan, S., Cohen, S.B. and Lapata, M., (2018a) Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. *CoRR*, [online] abs/1808.08745. Available at: http://arxiv.org/abs/1808.08745.
Narayan, S., Cohen, S.B. and Lapata, M., (2018b) Ranking Sentences for Extractive Summarization with Reinforcement Learning. *CoRR*, [online] abs/1802.08636. Available at: http://arxiv.org/abs/1802.08636.
Nogueira, R. and Cho, K., (2019) Passage Re-ranking with BERT. [online] pp.1–5. Available at: http://arxiv.org/abs/1901.04085.
Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y. and Dehak, N., (2019) Hierarchical Transformers for Long Document Classification. *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings*, pp.838–844.
Paulus, R., Xiong, C. and Socher, R., (2017) A Deep Reinforced Model for Abstractive Summarization. *CoRR*, [online] abs/1705.04304. Available at: http://arxiv.org/abs/1705.04304.
Pennington, J., Socher, R. and Manning, C.D., (2014) GloVe: Global Vectors for Word Representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. [online] pp.1532–1543. Available at: http://www.aclweb.org/anthology/D14-1162.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) Language Models are Unsupervised Multitask Learners.
Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., (2016) SQuad: 100,000+ questions for machine comprehension of text. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, ii, pp.2383–2392.
Richardson, M., Burges, C.J.C. and Renshaw, E., (2013) MCTest: A challenge dataset for the open-domain machine comprehension of text. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, October, pp.193–203.
Sang, E.F.T.K. and De Meulder, F., (2003) Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. [online] Available at: http://arxiv.org/abs/cs/0306050.
See, A., Liu, P.J. and Manning, C.D., (2017) Get To The Point: Summarization with Pointer-Generator Networks. *CoRR*, [online] abs/1704.04368. Available at: http://arxiv.org/abs/1704.04368.

Sennrich, R., Haddow, B. and Birch, A., (2016) Edinburgh Neural Machine Translation Systems for {WMT} 16. *CoRR*, [online] abs/1606.02891. Available at: http://arxiv.org/abs/1606.02891.
Seo, M., Kembhavi, A., Farhadi, A. and Hajishirzi, H., (2016) Bidirectional Attention Flow for Machine Comprehension. [online] pp.1–13. Available at: http://arxiv.org/abs/1611.01603.
Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.Y., (2019) MASS: Masked sequence to sequence pre-training for language generation. *36th International Conference on Machine Learning, ICML 2019*, 2019-June, pp.10384–10394.
UKRI, (2020a) *Can infected people without symptoms transmit coronavirus? - Coronavirus: the science explained - UKRI*. [online] Available at: https://coronavirusexplained.ukri.org/en/article/und0006/ [Accessed 3 Oct. 2020].
UKRI, (2020b) *Where did the new coronavirus come from? - Coronavirus: the science explained - UKRI*. [online] Available at: https://coronavirusexplained.ukri.org/en/article/cad0006/ [Accessed 4 Oct. 2020].
University of California, (2020) Study reveals how long COVID-19 remains infectious on cardboard, metal and plastic: People may acquire coronavirus through air and by touching contaminated surfaces. [online] Available at: https://www.sciencedaily.com/releases/2020/03/200320192755.htm [Accessed 3 Oct. 2020].
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems*. pp.5999–6009.
WHO, (2020) *WHO statement : Tobacco use and COVID-19*. [online] Available at: https://www.who.int/news-room/detail/11-05-2020-who-statement-tobacco-use-and-covid-19 [Accessed 4 Oct. 2020].
Williams, A., Nangia, N. and Bowman, S.R., (2018) A broad-coverage challenge corpus for sentence understanding through inference. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, pp.1112–1122.
Yang, P., Fang, H. and Lin, J., (2018) Anserini: Reproducible ranking baselines using lucene. *Journal of Data and Information Quality*, 104.
Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M. and Lin, J., (2019a) End-to-end open-domain question answering with BERTserini. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session*. pp.72–77.
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. V., (2019b) XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32NeurIPS, pp.1–18.
Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M. and Zhao, T., (2018) Neural Document Summarization by Jointly Learning to Score and Select Sentences. *CoRR*, [online] abs/1807.02305. Available at: http://arxiv.org/abs/1807.02305.

qa_dt_v2.xlsx

APPENDIX A:

df_ans_v2.xlsx

APPENDIX B:

APPENDIX C: Research plan

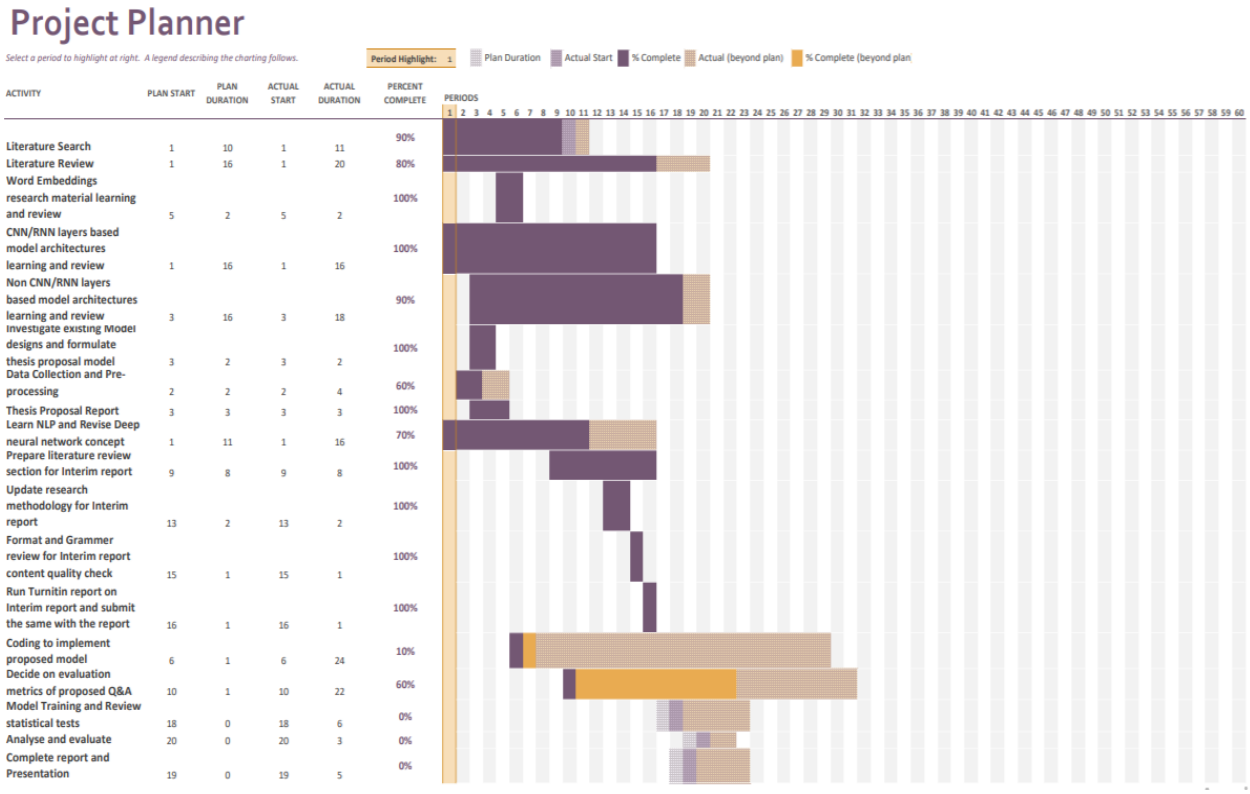

Figure 17: Project plan

Please note the following:

- Each “period” represents a week
- Week “1” is considered as the week in which work on approved project was started.

APPENDIX B: RESEARCH PROPOSAL

COVID-19 based dataset Kaggle Challenge TASK Solving Question and Answer Model Based on Novel Hierarchical Model Architecture Utilizing Existing State of Art Model Algorithms

Prashant Chadha
Student ID - 927706

Under the supervision of
SUVAJIT MUKHOPADHYAY

Research Proposal

Liverpool John Moores University - Master's in Data Science

JULY 2020

Abstract

The objective for my project is to solve Kaggle COVID-19 challenge Task query - "What is known about transmission, incubation, and environmental stability?" (Initiative, 2020). Corona Virus has disrupted life world over and this Kaggle challenge aims to aid in research about this virus by providing answers to task-based questions related to COVID 19 like the one mentioned above.

This task will be solved with the help of deep learning based "Attention" concept. The Attention mechanism in Deep Learning is based off concept of directing focus and pay greater attention to certain factors when processing the data. It's one of the foremost methodologies utilized extensively in Q&A model design. The hierarchical model design will be created using two different stages: first stage will consist of document ranking or classification in order to filter relevant documents while second stage would be Bi-Directional Attention Flow (BIDAF) network to get the answers to the task. The dataset includes a sizeable number of documents and to shortlist top documents that would consist answers to the task requirements, documents will be ranked using two approaches: dot product of TF-IDF values of documents as well as those of task question. The other approach will include using BERT model to classify the documents as relevant and non-relevant document with respect to the task query. The shortlisted documents will then be fed into the Question and Answering model to get the required answer span. This model will be implemented using both, BERT again as Question and Answering model and another bidirectional attention flow-based model. The two model architectural results based on evaluation metrics will be compared and the one with better scores will be selected.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Expansion**
ML	Machine Learning
AI	Artificial Intelligence
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
LSA	Latent Semantic Analysis
LDA	Latent Dirichlet allocation
Seq2Seq	Sequence to Sequence
DBQA	Document based Question and Answer
Q&A	Question and Answer
SQUAD	Stanford Question Answering Dataset

1. Background

Corona Virus, first found in a patient in Wuhan-China at the end of 2019, has since spread in more than 180 countries across the globe and resulted in both, the loss of life and economy within these countries. It was declared pandemic by WHO in January and after disrupting lives in China, Italy and Spain at the end of last year and early part of 2020, it has caused massive loss of lives in countries such as US, India, Brazil, UK, Germany, France and stalled economic activities across the globe and continue to spread.

Scientists across the globe have been studying the virus characteristics in order to suggest ways to reduce the spread, aid in development of therapeutics and vaccines. Since the virus is highly contagious and lethal to certain population, made evident by its impact across the globe, it's important that scientists can focus on relevant information shared by fellow colleagues by their documented literature to find answers that can help in design of possible virus spread mitigating steps as well as development of cure. The Kaggle challenge, COVID-19 Open Research Dataset Challenge (CORD-19), is an initiative to encourage data scientist community to develop an algorithm that can aid scientists by learning from extensive literature and then giving relevant answers to the task query.

The selected task query delves with detailing the important characteristics of this Virus such as:

- Range of incubation periods for the disease in humans (and how this varies across age and health status) and how long individuals are contagious, even after recovery.
- Prevalence of asymptomatic shedding and transmission (e.g., particularly children).
- Seasonality of transmission.
- Physical science of the coronavirus (e.g., charge distribution, adhesion to hydrophilic/phobic surfaces, environmental survival to inform decontamination efforts for affected areas and provide information about viral shedding).
- Persistence and stability on a multitude of substrates and sources (e.g., nasal discharge, sputum, urine, fecal matter, blood).

Complete list of expected characteristics to report can be accessed at the website (Initiative, 2020).

It's imperative that model designed is capable of learning from exhaustive number of detailed documents and can provide relevant answer span with respect to task query. It's for this requirement that bi-direction attention flow-based models are best suited for answering queries by retrieving relevant content from large corpus. The model architecture should also be able to shortlist or rank documents before retrieving the answer(s) as focusing on most relevant documents from large dataset corpus shared by Kaggle (Initiative, 2020) would aid in improving the performance as well as accuracy of model. Therefore, the decision to design a hierarchical model to answer the task query.

2. Problem Statement

The most extensively used concept now used in design of Q&A model is Attention Concept. As per the web article (Loye, 2019) on Attention mechanism: The Attention mechanism in Deep Learning is based off concept of directing focus and pay greater attention to certain factors when processing the data.

In broad terms, Attention is one component of a network's architecture, and oversees managing and quantifying the interdependence:

- Between the input and output elements (General Attention)
- Within the input elements (Self-Attention)

Attention was originally introduced as a solution to address the main issue surrounding sequence to sequence (Seq2Seq) models or Encoder-Decoder model, and to great success.

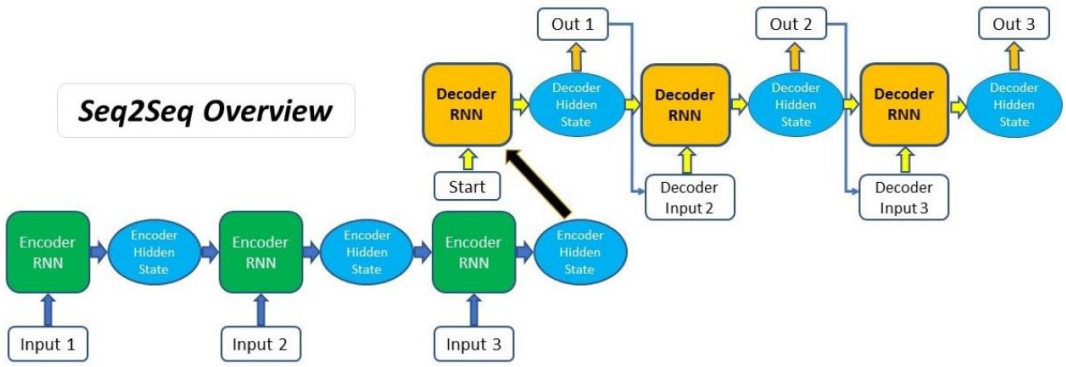

Figure 18: Seq2Seq Attention Process Flow in the Model (Loye, 2019)

The standard sequence-sequence model is generally unable to accurately process long input sequences, since only the last hidden state of the encoder RNN is used as the context vector for

the decoder. On the other hand, the Attention Mechanism directly addresses this issue as it retains and utilizes all the hidden states of the input sequence during the decoding process. It does this by creating a unique mapping between each time step of the decoder output to all the encoder hidden states. This means that for each output that the decoder makes, it has access to the entire input sequence and can selectively pick out specific elements from that sequence to produce the output.

There are two types of Attention principle-based models:

- Bahdanau Attention
- Luong Attention

One of the research work, "A Unified Model for Document-Based Question Answering Based on Human-Like Reading Strategy" (Li et al., 2018), authors utilized Attention principle to design unified model that contains three major encoding layers that are consistent to different steps of the reading strategy, including the basic encoder, combined encoder and hierarchical encoder. They focused on designing the model that could imitate human-like reading strategy for document-based Question and Answering (DBQA) task. As stated by authors, in the field of sentence pairs matching, there have been various deep neural network models proposed. Two levels of matching strategies are considered: the first is converting the whole source and target sentence into embedding vectors of latent semantic spaces respectively, and then calculating similarity score between them; the second is calculating the similarity score among all possible local positions of source and target sentences, and then summarizing the local scores into the final similarity score. Authors of this paper went with a novel approach of designing the model based on human-like reading strategy to tackle the DBQA problem, which could be conducted via the neural network.

The reading strategy involved first making a "summary" or "title" embedded vector of document in the first stage. As per the second step, authors incorporated the hidden representation of the title into the question, posing a limitation to the understanding of it and making the meaning closer to the document. Several methods, including deep learning models and simple computations, to combine both information were implemented by authors. Thirdly, a hierarchical RNN structure was employed to obtain the document level representation, equipped with the new question's encoding vector.

Latent Semantic analysis (LSA) and Latent Dirichlet allocation (LDA) were used to get an overall summary of a document. These methods of getting the summary focused on first and last sentences in the document.

The model was tested on English WikiQA dataset as well as Chinese DBQA dataset. It performed well on both datasets. The model architecture detailed in this paper helped develop insights about the hierarchical model design capable of filtering important content in the document and then use the same to answer respective query. However, the "summary" generation method will not work in dataset considered for thesis project as answer need to be formulated from multiple documents and so the first and last sentences will not reflect the summary of complete data to be considered

"Attention Is All You Need" (Vaswani et al., 2017), research paper introduced concept of Transformer. Authors designed model architecture utilizing only attention mechanism without use of CNN or RNN network. They wanted to design a self-attention based network that would avoid traditional RNN and CNN based model architecture bottlenecks at the time related to sequential computation and modeling of dependencies between text corpus without regard to their distance in the input or output sequences. Self-attention, mechanism that relates different portions of a single sequence in order to compute sequence representation, was achieved via implementation of a novel Transformer architecture consisting of stacked multi-layer Encoder and Decoder architecture. Encoder consists of stack of six identical layers. Each layer consisting of two sub layers: multi-head attention layers (8) and a feet forward neural network. Both sub layers surrounded by addition and normalization layer steps. Decoder design consisted of similar structure but has an additional Masked Multi-head attention layer. One of the multi-head attention layers is common between encoder and decoder. Linear and Soft-max layer follows the decoder, with output sequence generated by Soft-max layer. Utilizing multi-head attention layer architecture state of the art results comparable with existing language translation models at the time were achieved on same dataset at much faster pace. This paper helped in understanding concept of self-attention and its significance in learning relationship between texts separated over long distance in a sequence.

Question and Answer model Framework capable of retrieving answers from long documents was introduced by "Coarse to Fine Question Answering for Long Documents" (Choi et al., 2017) research work. Authors developed a hierarchical network comprising of coarse, fast

model for selecting relevant sentences and an expensive RNN based model to produce answer from the selected sentences. Sentence selection, treated as latent variable and trained together with answer generating model using re-inforcement learning, helped in selecting correct sentences as per context. Three types of mechanisms were considered for sentence selection model: Bag of words model, Chunking Bag of words model and Convolutional (CNN) model. Document summary was generated using the selected sentences using soft or deterministic attention as well as hard or stochastic attention. The summary was then fed to answer generating model. The answer generating model was designed using an encoder and decoder network based on Gated Recurrent Unit (GRU). Three types of learning approaches were used:

- Pipeline model learning, distant supervision wherein sentence selection model and answer generating model trained separately.
- Soft Attention learning, fully differentiable and optimized end-to-end with back propagation
- Hard Attention approach: optimized with Reinforce algorithm

State of the art results were achieved by this network compared with existing models at the time on same dataset and different learning strategies as well as sentence selection model performances were compared. Model architecture showcased great performance on datasets with long documents but was only explored and studied for answer retrieval from a single document, therefore, different strategy is needed for dataset and task requirement of thesis project. Nonetheless, the sentence selection mechanisms utilized in this work were enlightening as this approach inspired the document shortlisting approach being considered in the current thesis project.

Most of the initial Question and Answer as well as other natural language processing (NLP) task-based models utilized single direction, i.e., either Left sequence learning or right sequence learning model. These worked on short sentences, but the contextual learning was never fully realized over long sequences. “BI-DIRECTIONAL ATTENTION FLOW
FOR MACHINE COMPREHENSION” (Seo et al., 2016) introduced Bi-Directional Attention Flow (BIDAF) network, a multi-stage hierarchical process that represents the context at different levels of granularity and uses bi-directional attention flow mechanism to obtain a query-aware context representation without early summarization. BIDAF includes character-level, word-level, and contextual embeddings, and uses bi-directional attention flow to obtain a query-aware context representation.

Unlike few of other research works, author of this paper does not summarize the context paragraph into a fixed-size vector. Instead, the attention is computed for every time step, the attended vector at each time step, along with the representations from previous layers, can flow through to the subsequent modeling layer. This reduces the information loss caused by early summarization.

Second, authors used a memory-less attention mechanism. While they iteratively computed attention through time as in Bahdanau (Bahdanau et al., 2015) attention model, the attention at each time step is a function of only the query and the context paragraph at the current time step and does not directly depend on the attention at the previous time step. This forces the attention layer to focus on learning the attention between the query and the context, and enables the modeling layer to focus on learning the interaction within the query-aware context representation (the output of the attention layer)

Third, attention mechanisms in both directions are implemented, query-to-context and context-to-query, which provide complimentary information to each other. The model achieved state-of-the-art results in Stanford Question Answering Dataset (SQUAD) and CNN/DailyMail cloze test.

The model characteristics mentioned above made it ideal to be considered as one of the answer spans generating model in the hierarchical network to be designed to solve the task query

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (Devlin et al., 2019) research work introduced the state of the art and the most versatile model yet, capable of solving multiple NLP use cases including Question and Answer task. It can be incorporated using transfer learning, pretraining on known large dataset and then fine tuned on actual dataset. Authors introduced novel approach of using Masked language modeling (LML) that allowed deep bidirectional training by masking few random tokens and hence avoiding trivial responses from model. Utilizes bidirectional transformer network wherein encoder reads the entire sequence of words at once, allowing the model to learn the context of a word based on all its surrounding words. Two special tokens, [CLS]: inserted at the beginning of the first sentence and [SEP]: inserted at end of each sentence are used for data processing before sequence is fed as input. The CLS token is the classification token wherein after training it contains a vector that gives the information of the classification related downstream tasks. In case of question and answer task, this token would represent either the start and end span points, or, the document classification value based on its relevance to the query. The SEP token

separates two sentences or sentence and query pair that are input to the model after use of embeddings. Model architecture consists of bidirectional multi-attention layers encoder and decoder architecture as introduced by research work (Vaswani et al., 2017). BERT model is utilized both as document relevance ranker/classifier as well as answer span generation model in the current thesis project.

3. Research Questions

Following research questions are formulated as per the literature review and task solving strategy requirements:

- What's the most suited evaluation metrics to assess model performance taking into consideration answers are not explicitly available?
- What's the best granularity level (document, paragraph) to be considered as input to model?
- What's the ideal score to be considered while using TD-IDF values of query and data considered to have the answer?
- Which hierarchical model architecture out of the options being considered gives the best result?

4. Aim and Objectives

The main aim of this research is to develop a hierarchical network architecture capable of answering the Kaggle task query based on learning of content in extensive dataset. Therefore, goal of this research is to explore and demonstrate the performance of novel two stage model architecture options consisting of existing state of the art NLP task-based models.

The research objectives are formulated based on the aim of this study which is as follows:

- To develop robust and efficient question and answer model capable of learning content from exhaustive list of research documents and answer the task query
- To suggest the best performing hierarchical model network with respect to answers generated by model network options being considered
- To develop evaluation metric strategy most suited to judge the performance of model
- To identify appropriate granularity level to be considered for input sequence

5. Significance of the Study

The research study would aid in development of model capable of answering Corona Virus related task query, given huge repository of research papers developed by scientists studying the Virus. This would help the scientists in getting required details about virus in a much more efficient and faster way. Thereby, enabling scientists suggest virus spread mitigation steps, development of therapeutic treatment.

6. Scope of the study

As per the fixed time frame schedule, below limitations are set on this research to ensure timely completion of work:

- The dataset considered is restricted to the pdf files shared over Kaggle website only.
- The objective of this research is restricted to solving the task query mentioned above. The task query will be represented as list of sub-questions, focusing on specific Corona virus details asked by the task query.
- Evaluation technique(s) formulation that suits for assessment of the proposed model design is also part of scope of this study.

7. Research Methodology

7.1 Dataset Description

Dataset (Initiative, 2020) consists of pdf and xml documents in jason format. The scope of this research will be restricted on the pdf documents in jason format. Following is the breakup of folders and documents within them:

- biorxiv_medrxiv:
  - PDF - 1342 full text
- comm_use_subset:
  - PDF - 9365 full text
  - PMC - 8995 full text
- custom_license:
  - PDF - 23152 full text
  - PMC - 4773 full text
- noncomm_use_subset:
  - PDF - 2377 full text
  - PMC - 2093 full text

After looking into limited data set, in python notebook, focusing on documents in folder "biorxiv_medrxiv", following observations have been made:

- Each file has data within seven different dictionaries. The dictionaries are as follows:
  - paper_id
  - metadata
  - abstract

- body_text
- bib_entries
- ref_entries
- back_matter

- Relevant research content is within dictionary "body_text" and so will be considered for extraction of respective answers with respect to task query.

7.2 Data Pre-Processing and Transformation

Data that needs to be processed consists of research work on Corona Virus by authors of respective papers. The "body_text" dictionary within each document consist of research data and as part of data pre-processing, list consisting of "body_text" data from all documents within one of the folders (there are four folders as mentioned above) is created. Therefore, in all four such lists would be created. The name of respective documents is saved in another list in same order.

The concept behind creating four such lists is that it becomes easier to track the respective folder containing the documents relevant to task queries as per results of the document shortlisting models.

Once the list is created, following are the steps considered as part of data transformation before being used as input to the respective models:

Preprocess data (Query as well as data within the list) for the TM-IDF stage:

- Converting of all characters within the query as well as respective document data within the list to lower case.
- Use "RegEX" function to make sure only alpha numeric characters or valid web links exist in data using custom filter condition
- Remove multiple spaces between sentences by replacing such whitespaces with single whitespace.
- Use "TreebankWordTokenizer" function, available within python pacakge, to split sentences into individual words or tokens.
- Remove stop words (most frequent words that are non-essential for query-sentence contextual relationship task) from the above list with tokens using python function, "stopwords".
- Use "stemmer" function to stem the tokens.
- Join the resulting tokens with single space to form sentences.

Tokenization, data pre-processing for BERT and the other bi-directional model will be as per their architectural requirements.

7.3 Models

The high-level architectural diagram of hierarchical network being proposed to provide answer to the task queries is as follows:

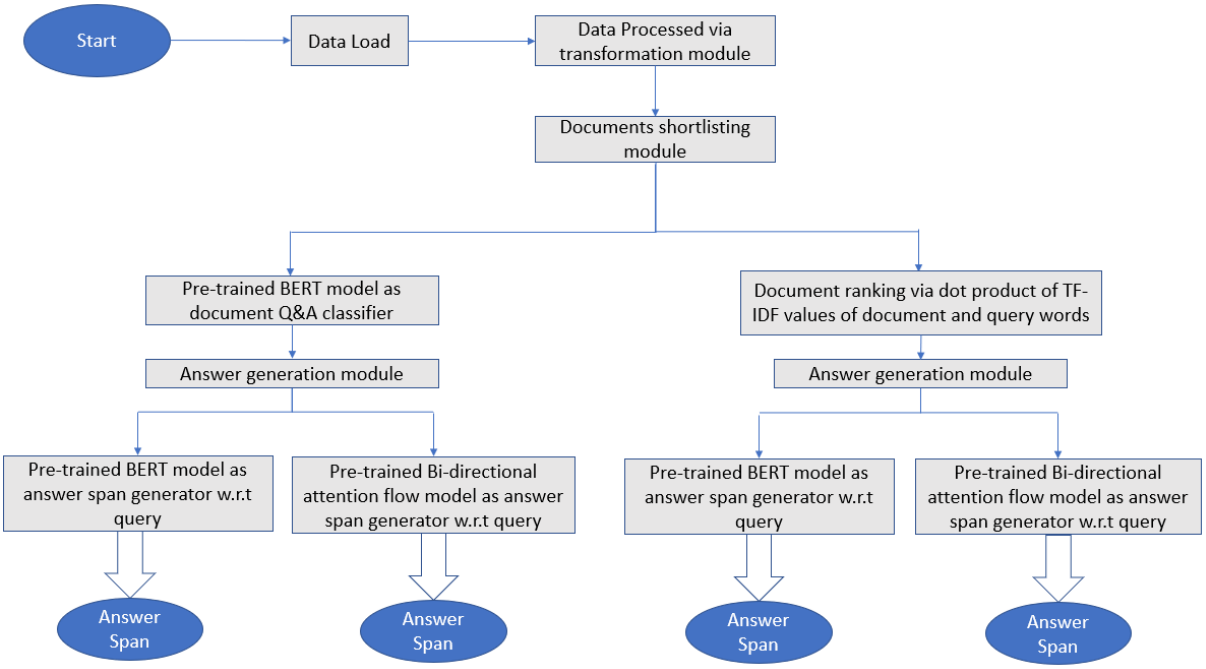

Figure 19: Hierarchical model process flow diagram

7.3.1 TF-IDF document ranking model

Term frequency–inverse document frequency (TFIDF) is a numerical statistic intended to reflect the importance of a word with respect to a document in a collection or corpus. It has often been used as a weighting factor in searches of information retrieval, text mining, and user modeling. The TF–IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. This helps to adjust for the fact that generally some words appear more frequently in text corpus.

The transformed data and query are transformed into respective TF-IDF word-importance value matrix using NLTK package function for TF-IDF vector value generation, “TfidfVectorizer”.
These vector values are then converted into python dataframe, wherein rows represent TF-IDF values of words in respective documents and columns are words that exist in the documents within the list.
Dot product between the TF-IDF values in the dataframe and query TF-IDF values is taken. The resulting score for each document in the dataframe is then taken by summing up the values over each row, representing the combined relevance of words within respective documents with respect to the query. Top few documents are then selected for consideration during the answer generation stage. The apt number of documents based on ideal cut-off score is one of the research questions.

7.3.2 BERT model

Bidirectional encoder representations from transformers, BERT (Devlin et al., 2019) model uses bidirectional attention to extract deep relationship between the query-sentence pair with help of novel approach, Masked Language Model (MLM). Since its inception, model has proven to give state of the art results in various NLP tasks such as language translation, sentiment analysis, named entity recognition (NER) as well as question and answer (Q&A) classification and answer span generation with respect to query.

The model architecture is based on self-attention based multi-layer encoder-decoder architecture (Vaswani et al., 2017). BERTBase (L=12, H=768, A=12, Total Parameters=110M) (Devlin et al., 2019), with L being the number of layers or transformer blocks, H is the hidden vector size and A is the number of attention heads, will be first used and based on performance as well as time, BERTLarge variant may be used.

Tokenization and Special Tokens

The word tokenization used by BERT is “Wordpiece” tokenization that will be used for tokenizing sequence consisting of both query and passage (document or paragraph within document based on granularity). “Wordpiece” embeddings consist of vocabulary of 30,000 tokens. In case a token in the input is not present, the word is broken into pieces/individual characters and are mapped with ID from the embeddings library. The maximum acceptable sequence length is 512 tokens and it’s a hyperparameter that can be tuned. In case the sequence is longer, its truncated and if the size of a given input sequence is less than the set size, its padded using special token [PAD]

[CLS] and [SEP] are the two special tokens utilized by BERT. [CLS] is the first token of input sequence, comprising of query and document data stream (document or paragraph). [SEP] token separates the query and the data sequence from documents. The role of [CLS] token is dependent on the final output expected from model. In case of this project, [CLS] token will be used as follows:

- Document classifier or ranking indicator in the document shortlisting task. The aggregated [CLS] token received from the final output layer will classify document as relevant or non-relevant with respect to query.
- The start and stop indicators to highlight the answer span in the case of BERT used as answer span generation with respect to query.

Model training

Model will be pre-trained on English Wikipedia (2,500M words) considering only the text passages, ignoring headers, tables and lists. Fine tuning will be performed on the project dataset. Following are the important and novel pre-training approaches utilized for BERT model:

Masked LM (MLM)

BERT utilizes deep bidirectional attention with help of novel masked LM technique. This involves masking of some percentage of tokens and then let model predict these tokens for it to learn deep bidirectional representation within sequence. The strategy followed is as proposed by author (Devlin et al., 2019), i.e., 15% of all Wordpiece tokens in each sequence are masked at random and only these masked tokens are predicted.
Within this 15% masked tokens, 80% tokens are masked, 10% tokens are replaced by random tokens while rest of 10% are left as original tokens. The 15% tokens are selected randomly.

Next Sentence Prediction (NSP)

This process helps in model understanding the relationship between two sentences. As part of this pre-training task, two sentences, sentence A and B respectively are selected from text corpus. 50% of the time, sentence B is the actual next sentence to following sentence A and is labelled as “IsNext”, while, for rest of 50% cases, sentence B is a random sentence from the corpus and is labelled as “NotNext”.

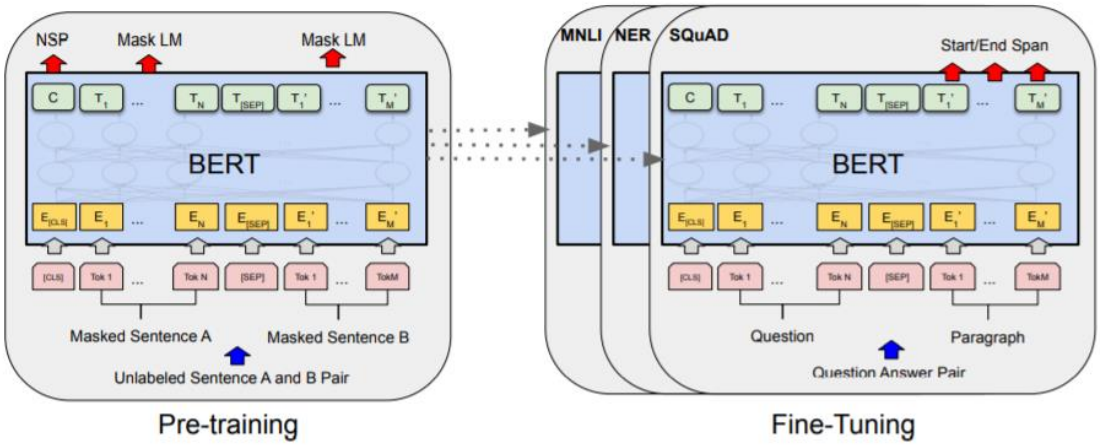

Figure 20: BERT pre-training and Fine-Tuning process (Devlin et al., 2019)

BERT Fine-tuning

BERT fine tuning is a straightforward process and will be carried out on project data set to get document ranking or classification during the document shortlisting task as well as get the answer span with respect to query during the answer span generation task.
The self-attention mechanism utilized by BERT to encode both the query and passage concatenated tokens simultaneously and hence, enables bidirectional cross section attention between the query and passage. This helps in both, identifying the relevance of passage(s) within the document as well as get the required answer span with the help of [CLS] tokens.

Evaluation strategy is part of research question and will be finalized as research goes forward.

7.3.3 Bi-Directional Attention Flow (BIDAF) network for answer span generation

This is a hierarchical multi-stage architecture, introduced in “BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION” (Seo et al., 2016), for modeling the representations of the context paragraph at different levels of granularity, as shown in figure 4. BIDAF network includes character-level, word-level, and contextual embeddings, and uses bi-directional attention flow to obtain a query-aware context representation.
The advantages of BIDAF network over traditional network are:

- Attention layer is not used to summarize the context paragraph into a fixed-size vector. Instead, it is computed for every time step, and the attended vector at each time step, along with the representations from previous layers, flow through to the subsequent modeling layer. This reduces the information loss caused by early summarization.
- Memory-less attention mechanism: As per this mechanism, attention is iteratively computed through time as in the Bahdanau attention model (Bahdanau et al., 2015). However, the attention at each time step is a function of only the query and the context paragraph at the current time step and does not directly depend on the attention at the previous time step. This leads to the division of responsibilities between the attention layer and the modeling layer.
  The attention layer focuses on learning the attention between the query and the context and enables the modeling layer to focus on learning the interaction within the query-aware context representation (the output of the attention layer).

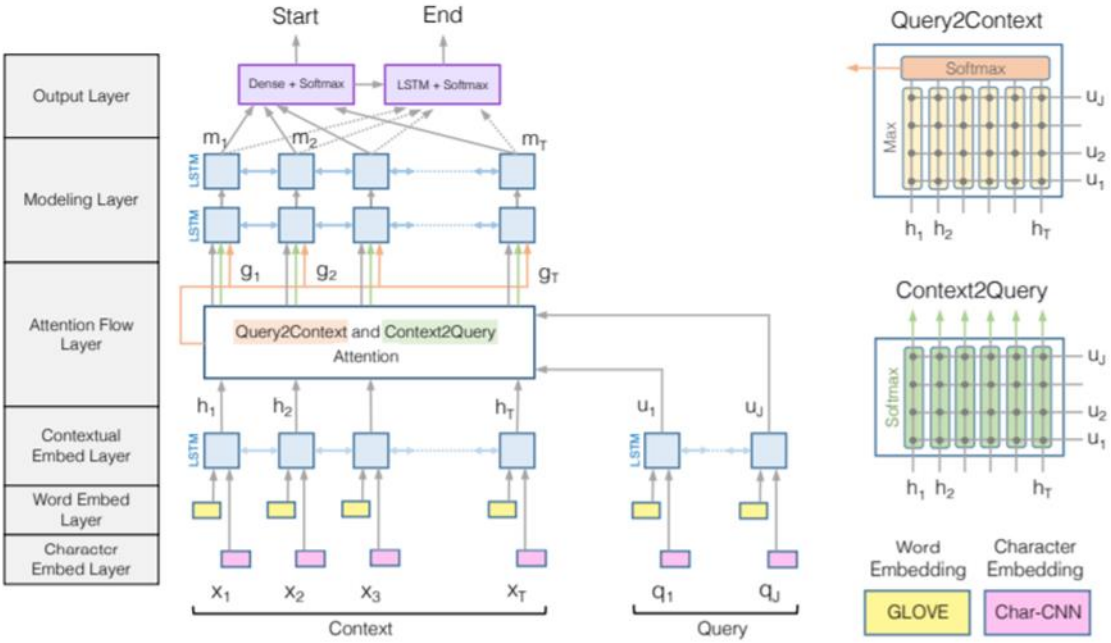

Figure 21: Bidirectional Attention Flow Model (Seo et al., 2016)

The model architecture that will be used consists of following layers:

- Character Embedding Layer maps each word to a vector space using character-level CNNs.
- Word Embedding Layer maps each word to a vector space using a pre-trained word embedding model.
- Contextual Embedding Layer utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context.
- Attention Flow Layer couples the query and context vectors and produces a set of query aware feature vectors for each word in the context.
- Modeling Layer employs a Recurrent Neural Network to scan the context.
- Output Layer provides an answer to the query. The answer would be span within the context of the query.

Tokenization and Embeddings

The query and passage (paragraph or complete document) tokenization will be performed using regular expression-based word tokenizer (PTB tokenizer). Pre-trained word vectors such as GloVe (HANSON ER, 1971), will be used for word embedding of each token. Different word embedding could be considered based on further research.

Model Training and Evaluation

The training loss (to be minimized) objective defined for this model is the sum of the negative log probabilities of the true start and end indices by the predicted distributions, averaged over all examples.
Evaluation strategy is part of research question and will be finalized as research goes forward.

8. Requirements / resources

8.1 Hardware Requirements

As per research work studied so far related to implementation of selected models, following minimum hardware configuration would be required:

Processor: Intel Xeon E5-2620 v4 CPU (2.10GHz)

Memory: 8 GB LPDDR3 2133MHz

GPU: Tesla P40 GPU or TITAN X GPU

8.2 Software Requirements

The Model design would require to be implemented in the latest Python framework. The implementation involves usage of open source libraries such as Pandas and NumPy for data processing and manipulation, Matplotlib and Seaborn for EDA, NLTK and TensorFlow packages for NLP and neural network functions and models

9. Research Plan

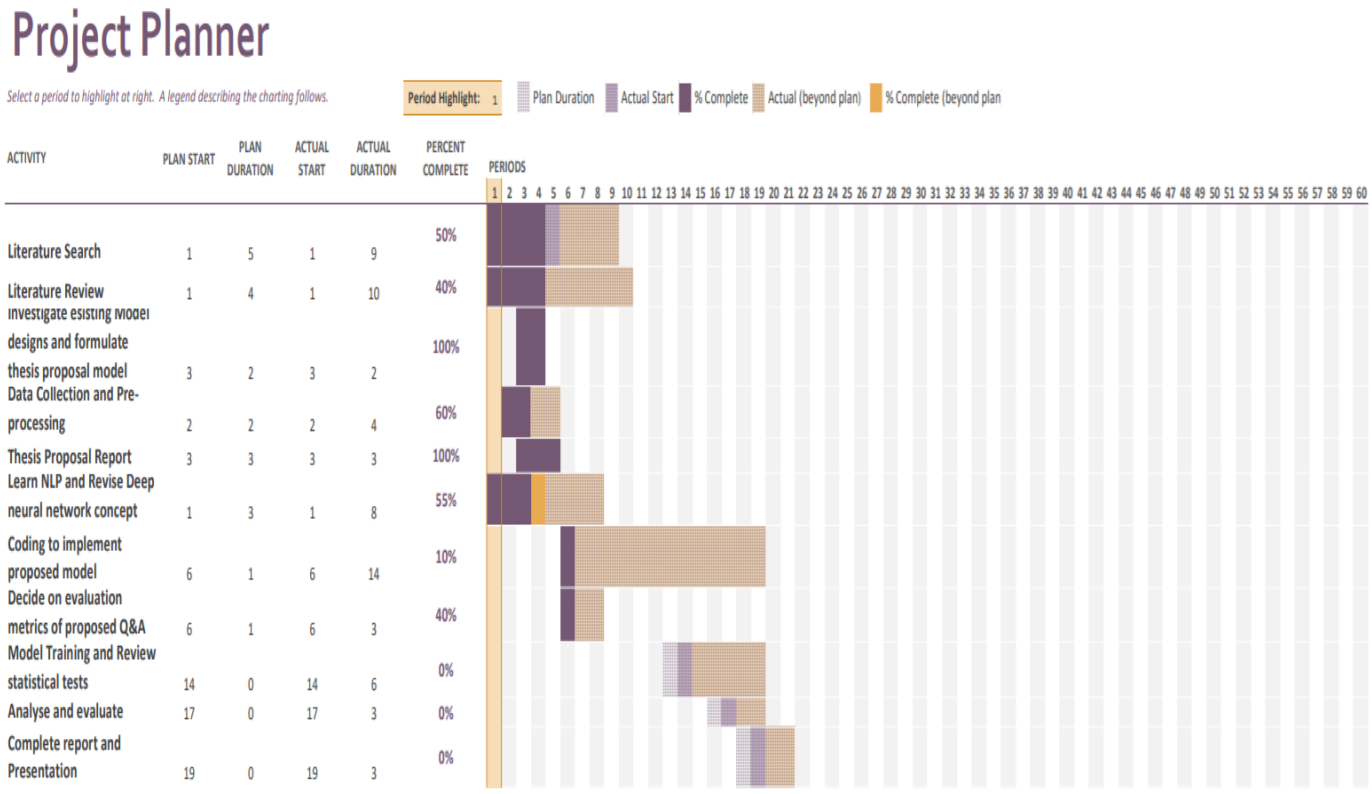

Figure 22: Research plan

Please note the following:

- Each “period” represents a week
- Week “1” is considered as the week in which work on approved project was started.

References

ACOG, (2020) *Coronavirus (COVID-19), Pregnancy, and Breastfeeding | ACOG*. [online] Available at: https://www.acog.org/patient-resources/faqs/pregnancy/coronavirus-pregnancy-and-breastfeeding#How does COVID19 affect pregnant women [Accessed 4 Oct. 2020].

Allen Institute For AI, Goldbloom, A., Lin, P., Mooney, P., Carissa, S., Kohlmeier, S., Devrishi, Bozsolik, T. and Hamner, B., (2020) *COVID-19 Open Research Dataset Challenge (CORD-19) | Kaggle*. [online] Available at: https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=568 [Accessed 4 Sep. 2020].

Bahdanau, D., Cho, K.H. and Bengio, Y., (2015) Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp.1–15.

Bengio, Y., (2003) A Neural Probabilistic Language Model Yoshua. *Fullerenes Nanotubes and Carbon Nanostructures*, 268, pp.465–470.

Çelikyilmaz, A., Bosselut, A., He, X. and Choi, Y., (2018) Deep Communicating Agents for Abstractive Summarization. *CoRR*, [online] abs/1803.10357. Available at: http://arxiv.org/abs/1803.10357.

Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A. and Berant, J., (2017) Coarse-to-fine question answering for long documents. In: *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. [online] pp.209–220. Available at: http://arxiv.org/abs/1611.01839.

Davies, N.G., Klepac, P., Liu, Y., Prem, K., Jit, M., Pearson, C.A.B., Quilty, B.J., Kucharski, A.J., Gibbs, H., Clifford, S., Gimma, A., van Zandvoort, K., Munday, J.D., Diamond, C., Edmunds, W.J., Houben, R.M.G.J., Hellewell, J., Russell, T.W., Abbott, S., Funk, S., Bosse, N.I., Sun, Y.F., Flasche, S., Rosello, A., Jarvis, C.I. and Eggo, R.M., (2020) Age-dependent effects in the transmission and control of COVID-19 epidemics. *Nature Medicine*, [online] 268, pp.1205–1211. Available at: https://doi.org/10.1038/s41591-020-0962-9 [Accessed 3 Oct. 2020].

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. pp.4171–4186.

Dietz, L., Verma, M., Radlinski, F. and Craswell, N., (2012) TREC Complex Answer Retrieval Overview. pp.1–13.

Dinan, E., Logacheva, V., Malykh, V., Miller, A.H., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., Prabhumoye, S., Black, A.W., Rudnicky, A.I., Williams, J., Pineau, J., Burtsev, M.S. and Weston, J., (2019) The Second Conversational Intelligence Challenge (ConvAI2). *CoRR*, [online] abs/1902.00098. Available at: http://arxiv.org/abs/1902.00098.

Dirk, (2020) *Anserini+BERT-SQuAD for Semantic Corpus Search | Kaggle*. [online] Available at: https://www.kaggle.com/dirktheeng/anserini-bert-squad-for-semantic-corpus-search/notebook [Accessed 25 Aug. 2020].

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.W., (2019) Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32NeurIPS.

Durrett, G., Berg-Kirkpatrick, T. and Klein, D., (2016) Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. *CoRR*, [online] abs/1603.08887. Available at: http://arxiv.org/abs/1603.08887.

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J. and Auli, M., (2019) {ELI}5: Long Form Question Answering. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. [online] Florence, Italy: Association for Computational Linguistics, pp.3558–3567. Available at: https://www.aclweb.org/anthology/P19-1346.

Forster, P., Forster, L., Renfrew, C. and Forster, M., (2020) How genomic epidemiology is tracking the spread of COVID-19 locally and globally. *Proceedings of the National Academy of Sciences of the United States of America*, [online] 11717, pp.9241–9243. Available at: https://cen.acs.org/biological-chemistry/genomics/genomic-epidemiology-tracking-spread-COVID/98/i17 [Accessed 4 Oct. 2020].

Group, S.H.C.I., (2015) Daemo : a Self-Governed Crowdsourcing Marketplace. pp.2–3.

HANSON ER, (1971) MUSICASSETTE INTERCHANGEABILITY. THE FACTS BEHIND THE FACTS. *AES: Journal of the Audio Engineering Society*, 195, pp.417–425.

Harvard Health Publishing, (2020) *Preventing the spread of the coronavirus - Harvard Health*. *Harvard Health publishing*. Available at: https://www.health.harvard.edu/diseases-and-conditions/preventing-the-spread-of-the-coronavirus [Accessed 4 Oct. 2020].

Hermann, K.M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M. and Blunsom, P., (2015) Teaching Machines to Read and Comprehend. *CoRR*, [online] abs/1506.03340. Available at: http://arxiv.org/abs/1506.03340.
Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M. and Berthelot, D., (2015) : A Novel Large-scale Language Understanding Task over Wikipedia.
Huang, Y., Sun, M. and Sui, Y., (2020) How Digital Contact Tracing Slowed Covid-19 in East Asia. *Harvard Business Review Digital Article*, [online] pp.1–8. Available at: https://hbr.org/2020/04/how-digital-contact-tracing-slowed-covid-19-in-east-asia [Accessed 5 Oct. 2020].
Initiative, A.I. for A. in partnership with the C.Z., (2020) COVID-19 Open Research Dataset Challenge (CORD-19). *Semantic Scholar*. [online] Available at: https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=568 [Accessed 6 Jul. 2020].
Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L., (2020) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. pp.7871–7880.
Li, W., Li, W. and Wu, Y., (2018) A unified model for document-based question answering based on human-like reading strategy. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. [online] pp.604–611. Available at: www.aaai.org.
Liu, Y., (2019) Fine-tune BERT for Extractive Summarization. [online] Available at: http://arxiv.org/abs/1903.10318.
Liu, Y. and Lapata, M., (2020) Text summarization with pretrained encoders. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp.3730–3740.
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. [online] 1. Available at: http://arxiv.org/abs/1907.11692.
Loye, G., (2019) Attention mechanism. *FLOYDHUB*, [online] pp.0–16. Available at: https://blog.floydhub.com/attention-mechanism/ [Accessed 6 Jul. 2020].
Luong, M.T., Socher, R. and Manning, C.D., (2013) Better word representations with recursive neural networks for morphology. *CoNLL 2013 - 17th Conference on Computational Natural Language Learning, Proceedings*, pp.104–113.
Mikolov, T., (n.d.) Language Models for Automatic Speech. 4.
Mikolov, T., Chen, K., Corrado, G. and Dean, J., (2013) Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp.1–12.
Mikolov, T., Deoras, A., Povey, D., Burget, L. and Černocký, J., (2011) Strategies for training large scale neural network language models. *2011 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2011, Proceedings*, pp.196–201.
Mikolov, T., Jiri, K., Burget, L., Glembek, O. and Cernocky, J. 'Honza', (n.d.) NEURAL NETWORK BASED LANGUAGE MODELS FOR HIGHLY INFLECTIVE LANGUAGES ˇ Tom ´ a s ˇ Mikolov , Ji ˇ r ´ ı Kopeck ´ y , Luk ´ a s ˇ Burget , Ond ˇ rej Glembek and Jan " Honza " Cernock ´ y Speech @ FIT , Faculty of Information Technology , Brno University of T. pp.2–5.
Narayan, S., Cohen, S.B. and Lapata, M., (2018a) Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. *CoRR*, [online] abs/1808.08745. Available at: http://arxiv.org/abs/1808.08745.
Narayan, S., Cohen, S.B. and Lapata, M., (2018b) Ranking Sentences for Extractive Summarization with Reinforcement Learning. *CoRR*, [online] abs/1802.08636. Available at: http://arxiv.org/abs/1802.08636.
Nogueira, R. and Cho, K., (2019) Passage Re-ranking with BERT. [online] pp.1–5. Available at: http://arxiv.org/abs/1901.04085.
Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y. and Dehak, N., (2019) Hierarchical Transformers for Long Document Classification. *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings*, pp.838–844.
Paulus, R., Xiong, C. and Socher, R., (2017) A Deep Reinforced Model for Abstractive Summarization. *CoRR*, [online] abs/1705.04304. Available at: http://arxiv.org/abs/1705.04304.
Pennington, J., Socher, R. and Manning, C.D., (2014) GloVe: Global Vectors for Word Representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. [online] pp.1532–1543. Available at: http://www.aclweb.org/anthology/D14-1162.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) Language Models

are Unsupervised Multitask Learners.
Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., (2016) SQuad: 100,000+ questions for machine comprehension of text. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, ii, pp.2383–2392.
Richardson, M., Burges, C.J.C. and Renshaw, E., (2013) MCTest: A challenge dataset for the open-domain machine comprehension of text. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, October, pp.193–203.
Sang, E.F.T.K. and De Meulder, F., (2003) Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. [online] Available at: http://arxiv.org/abs/cs/0306050.
See, A., Liu, P.J. and Manning, C.D., (2017) Get To The Point: Summarization with Pointer-Generator Networks. *CoRR*, [online] abs/1704.04368. Available at: http://arxiv.org/abs/1704.04368.
Sennrich, R., Haddow, B. and Birch, A., (2016) Edinburgh Neural Machine Translation Systems for {WMT} 16. *CoRR*, [online] abs/1606.02891. Available at: http://arxiv.org/abs/1606.02891.
Seo, M., Kembhavi, A., Farhadi, A. and Hajishirzi, H., (2016) Bidirectional Attention Flow for Machine Comprehension. [online] pp.1–13. Available at: http://arxiv.org/abs/1611.01603.
Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.Y., (2019) MASS: Masked sequence to sequence pre-training for language generation. *36th International Conference on Machine Learning, ICML 2019*, 2019-June, pp.10384–10394.
UKRI, (2020a) *Can infected people without symptoms transmit coronavirus? - Coronavirus: the science explained - UKRI*. [online] Available at: https://coronavirusexplained.ukri.org/en/article/und0006/ [Accessed 3 Oct. 2020].
UKRI, (2020b) *Where did the new coronavirus come from? - Coronavirus: the science explained - UKRI*. [online] Available at: https://coronavirusexplained.ukri.org/en/article/cad0006/ [Accessed 4 Oct. 2020].
University of California, (2020) Study reveals how long COVID-19 remains infectious on cardboard, metal and plastic: People may acquire coronavirus through air and by touching contaminated surfaces. [online] Available at: https://www.sciencedaily.com/releases/2020/03/200320192755.htm [Accessed 3 Oct. 2020].
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems*. pp.5999–6009.
WHO, (2020) *WHO statement : Tobacco use and COVID-19*. [online] Available at: https://www.who.int/news-room/detail/11-05-2020-who-statement-tobacco-use-and-covid-19 [Accessed 4 Oct. 2020].
Williams, A., Nangia, N. and Bowman, S.R., (2018) A broad-coverage challenge corpus for sentence understanding through inference. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, pp.1112–1122.
Yang, P., Fang, H. and Lin, J., (2018) Anserini: Reproducible ranking baselines using lucene. *Journal of Data and Information Quality*, 104.
Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M. and Lin, J., (2019a) End-to-end open-domain question answering with BERTserini. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session*. pp.72–77.
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. V., (2019b) XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32NeurIPS, pp.1–18.
Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M. and Zhao, T., (2018) Neural Document Summarization by Jointly Learning to Score and Select Sentences. *CoRR*, [online] abs/1807.02305. Available at: http://arxiv.org/abs/1807.02305.