Question Answering Model with BERT

Final report

Hans Tiwari, Preetha Datta
1001998, 772587

This report is a part of the course
ELEC-E5550 Statistical Natural Language Processing.

Aalto University
27.04.2022

1 Introduction

There are often frequently asked question (FAQ) pages with various information on the web. With advances in natural language processing and text synthesis, there has been much work in creating automated question-answering models. A question-answering model refers to a comprehensive reading (often many paragraphs long) based on which certain questions are posed. In the context of NLP, a model then tries to answer the questions in the context of the given paragraphs. They are able to paraphrase answers in a generative manner, extract answers from a present paragraph, select the "correct" option given a list, and so on. Initial approaches to this problem involve a bag-of-words technique, which of course fails for larger datasets and more abstract questions.

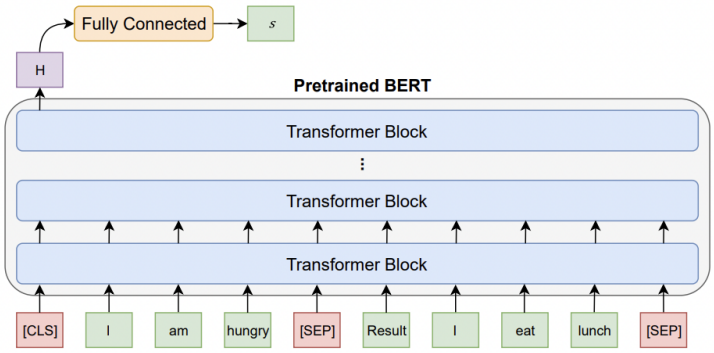


Figure 1: The BERT Model Architechture. Originally published in [1].

BERT (Bidirectional Encoder Representations from Transformers) is a state of the art language representation model designed to pre-train deep bidirectional representations[2]. It was developed in 2018 by researchers at Google AI Language and serves as a one-stop way to solve to some of the most important NLP challenges. This includes sentiment analysis and named entity recognition [3]. In the scope of this project, we will use BERT in order to create a question-answering model. BERT is known to have shown excellent results in some areas of natural language processing such as machine translation, etc. It will be interesting to understand and improve BERT's performance on a large dataset to create a question-answering model. We will adapt our approach in order to use the BERT architecture to its full potential [4].

In order to implement this model, we will be using the freely available BERT library provided by HuggingFace, and the SQuAD dataset in order to train our model. We will implement this project using Python. In the following sections, we will introduce BERT, its architecture, the SQuAD dataset in further detail and discuss how they fit into our group project.

2 Literature study

In this section, we describe the literature study we conducted before starting to work on our project task. Firstly, we define the concept of BERT including the model architecture and applications. Secondly, we discuss the SQuAD and how it can be used in the context of our project.

2.1 BERT

2.1.1 What is BERT?

Since its first appearance, BERT has gained academic renown. **B**idirectional **E**ncoder **R**epresentations from **T**ransformers is a model whose key feature is using a Masked Language Model (or MLM) pre-training in order to go over the unidirectionality previously applied in conventional transformer models [2]. BERT can be applied as a transfer learning model, which makes it versatile and applicable to many subdomains in the NLP problem space. It does this by using Transformers, which are models focused solely on attention mechanism and adept at understanding the relationship between different words [5]. A more detailed transformer architechture can be seen in Figure 2. BERT has been pretrained using self-supervision on a large corpus of English data. This implies that BERT was trained using raw data with very little human labelling input. This makes the model more powerful since it can be used to train further with publicly available information.

The BERT model was trained on BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables and headers)[6] .Since BERT relies so heavily on Transformers, which focus on reading the entire text sequence at once instead of a more traditional left to right approach, it would be more appropriate to think of BERT as non-directional instead of it being a bi-directional model [7].

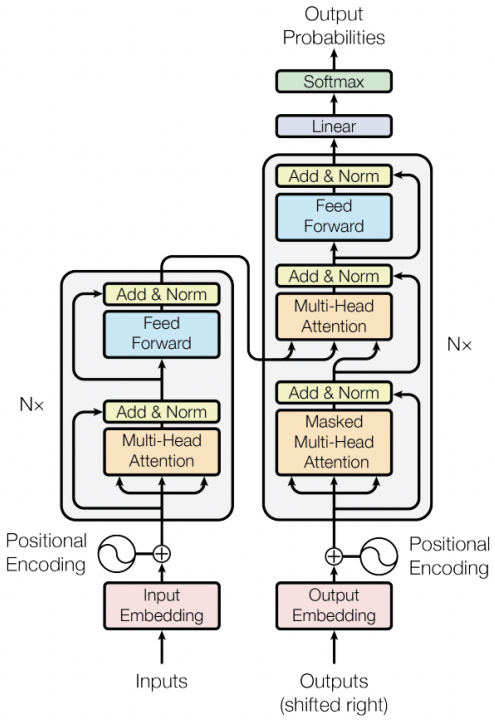


Figure 2: Transformer Architecture as presented by Vaswavi et all [5]. The BERT Architecture relies heavily on the groundwork laid out by the Transformer model.

2.1.2 BERT Architecture

BERT's model architecture is a multi-layer bidirectional Transformer encoder based on Transformer models as laid out in the paper by Vaswani et all [5]. The paper proposes two versions of BERT, one being the $BERT_{BASE}$ which has 12 Transformer Blocks, and 12 self-attention heads, with the total number of parameters coming to 110 Million. The $BERT_{LARGE}$ model has 24 Transformer blocks and a total of 16 Self-Attention heads, with a total number of parameters at 340 Million[2]. A more simple explanation of BERT is that it used a bidirectional transformer architecture stacking encoders from the original transformer on top of each other[8].

2.1.3 Pre-training BERT

BERT is pre-trained using two unsupervised tasks, the first one being the Masked Language Modelling task. We know that standard models are unidirectional, ie they can only be trained left-to-right or right-to-left. In order to train a deep bidirectional representation, a percentage of the input tokens are masked at random and the model then attempts to predict those masked tokens[2]. This is referred to as the Masked LM task. The model predicts only masked words in this model, in comparison to auto-encoders which reconstruct the entire input.

The second task in order to pre-train BERT is the Next Sentence Prediction or the NSP. Several NLP tasks rely on understanding how two sentences are related to one another [2]. Therefore, BERT uses a binarised next sentence prediction generated from a monolingual corpus.

2.1.4 Fine Tuning BERT

Fine-tuning is comparatively easier to implement as the self-attention mechanism in the Transformer does not restrict BERT to model downstream tasks. In BERT self-attention layer also helps in unifying the previous multiple tasks. For each task, we give task-specific input and output to BERT model and then fine-tune the overall parameters.[2]

Since BERT hypertunes all the parameters end-to-end, as compared to pre-training, the fine tuning is less expensive and takes less computational resources. It is this inexpensive fine-tuning will helps us use BERT model with limited computational resources, and makes it very relevant for our project.

2.2 BERT as QnA Model

Several studies have used BERT over many different datasets in order to create question-answer models. One such study in particular combines BERT with Answerini, an information-retrieval toolkit in order to create a unique question answering system [9]. The study proves that fine-tuning a BERT model with SQuAD shows high accuracy in finding out the spans of answers. Their model was finally able to achieve an F1 score of 46.1 and in fact, has been

deployed in production as well.

BERT has been integrated with some older methods to improve accuracy such as in ConvQA [10] which also takes in to account conversation history, ie, it attempts to improve context handling. The ConvQA model has three components – a ConvQA model, a history selection module, and a history modeling module.In practice, the history modeling module can be a mechanism inside the ConvQA model.This model achieved an impressive F1 score of 63.1.

Another study shows that BERT can be used to create a question answering system, even without the aid of a third party toolkit such as Answerini [11]. This study was also able to successfully surmount the bottlenecks of computation and data scarcity in order to create their question answering model, by matching a posed question to a pre-existing question and retrieving the correct answer from the dataset. The automatic system alone performs well and has been publicly hosted on Telegram for public testing. This particular study focused on cases where BERT was exploited to develop systems for real life applications. The paper used an external intent detection API as an evaluation tool and was able to achieve the correct intents on 84% of the cases.

Finally, we illustrate an example where BERT is used to train a question answering model on a specific topic – BioBERT is a domain-specific language representation model pre-trained on large-scale biomedical corpora[12]. BioBERT is far able to outperform BERT, after being trained on biomedical corpora (2.80% F1 score improvement). This shows that the baseline BERT model can very much be improved if a specific topic is so required.

2.3 SQuAD

The Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset consisting of 100,000+ questions posed by crowdworkers on a set of Wikipedia articles[13]. The answer to each question is a text segment or a corresponding reading passage, or the question might not be answerable. Top 10000 articles were retreived from Wikipedia's internal page rank in order to create this dataset. From this subset, 536 articles were sampled at random and individual paragraphs were extracted, after removing images, figures etc to give overall 23215 paragraphs over a vast range of topics.

The annotation for the dataset was covered by Mechanical Turk workers. For each selected paragraph, the workers were asked to come up with and answer 5 questions on the content of the paragraph. They were provided a text field to type their question, and they could highlight the answers in the paragraph [14].

2.4 SQuAD as Dataset for QnA Models

An interesting case where both BERT and SQuAD have been used in tandem is the paper presented by Mozannar et all. which describes a Neural Model for Question and Answering

in Arabic. This paper uses a machine translated version of SQuAD using the Google Translate neural machine translation (NMT) API as well as a pre-trained BERT model to create SOQAL. It is composed of three modules; Namely, a document retriever, a machine reading comprehension module, and an answer ranking module [15]. SOQAL was ultimately able to achieve an F1 score of 61.3, which is impressive considering the shortage of data in Arabic as compared to languages such as English. This paper is illustrative of how the SQuAD dataset can be used in creative ways to solve unique problem.

SQuAD dataset has been used extensively to create question answering models as well to augment specific datasets. A robust QA dataset was presented by Moller et all. to specifically create a Coronavirus answering model. The annotations were created in SQuAD style fashion where annotators mark text as answers and formulate corresponding questions. [16]. In this case, since the model specialised in answering information about Covid-19, the authors also trained their model on a COVID specific dataset, curated by human supervision. 147 scientific articles were selected mostly related to COVID-19 from the CORD-19 collection to be annotated by experts. A pre-trained BERT model on this dataset finally gave an F1 score of 59.53. This is a good example of how the SQuAD dataset augments question answering tasks.

2.5 BIDAF Model

We have also implemented a pre-trained Bidirectional Attention Flow model for comparison. We will now describe the BIDAF model briefly, and highlight why it is a good fit for question answering models.

The BIDAF model was originally intended for Machine Comprehension tasks in NLP, but was soon adapted for Question Answering. The Bi-Directional Attention Flow (BIDAF) network is a hierarchical multi-stage architecture for modeling the representations of the context paragraph at different levels of granularity and includes character-level, word-level, and contextual embeddings, using bi-directional attention flow to obtain a query-aware context representation. [17]. BIDAF is a closed-domain, which means it does not rely on pre-exiasting knowledge to solve a problem; rather the BIDAF models requires a context along with a question in order to be able to answer it. BIDAF is also extractive in nature which implies that it answers a query with a substring of the context that is the closest to the query. The BIDAF model consists of 6 layers which can be described as follows :

1. **Character Embedding Layer**: which uses character level CNN's to map every word to a vector

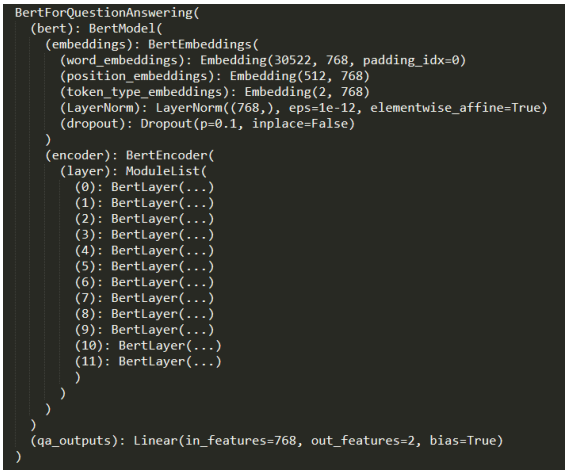
2. **Word Embedding Layer**: uses a pre-trained model to achieve mapping of each word to a vector space

3. **Contextual Embedding Layer**: refines embeddings of the words of the query as well as the context

4. **Attention Flow Layer**: outputs a set of query aware feature vectors for each word

5. **Modeling Layer**: uses an RNN to go through the context
6. **Output Layer**: produces a final answer to the context

3 Methods



(a) Overall Architecture

(b) Detailed Architecture

Figure 3: BERT Architecture: Further Details

This section describes the methods and tools used in our project. The goal of our project is to train a model that is concisely able to answer a given question with the context of a paragraph of text. We have implemented **four** different models to create a baseline and comparisons. Our first model is a BERT Base Uncased model; Secondly we use the SQuAD dataset to train the BERT based uncased model and are able to control hyperparameters to improve performance. Next, we implement a pretrained BERT model on the entire SQUAD dataset. Finally we implement a pre-trained BIDAF model to create a question-answering model. We will now primarily go into detail about the technical stacks and the give a brief description of the different models used in the project.

3.1 BERT Base Uncased Model

This could be considered the "baseline" model for our project. BERT Base Uncased is a pre-trained model on the English language using a masked language modeling (MLM) objective. This model is uncased which means it does not make a difference between english and English [6]. Since this is BERT Base, this particular madel has 12 encoder blocks stacked one on top of another, as opposed to 24 in BERT Large. In order to implement this model, we simply imported it from the `Transformer` library on Python and used the the fast BERT tokenisers from the `Transformer` library in order to tokenize our input.

3.2 Fine-Tuned BERT Model

We then trained the Bert Base Uncased model on the SQuAD v2 dataset. Our hyperparameters for training were as follows – We used **AdamW** (Adam with decoupled weight decay) optimizer which is a stochastic optimizer that avoids overfitting by changing the typical weight decay in Adam [18]. This is expecially useful in these cases, where the model has high complexity. Our learning rate was **0.005**. Batch size for the training was set to **8**. Finally, we ran **2 epochs** on the entire SQuAD dataset mostly due to computational restraints – each epoch took approximately **3 hours** to train on a GPU. Training for this model was implemented primarily on Google Colab. The BERT Model architecture can be seen in figure 3. Figure 3a shows the overall architecture of the BERT base architecture. It can be seen that there are twelve encoding layers in the overall architecture. Figure 3b shows a more detailed architecture of each layer in our BERT model.

3.3 Pretrained BERT Model on SQUAD

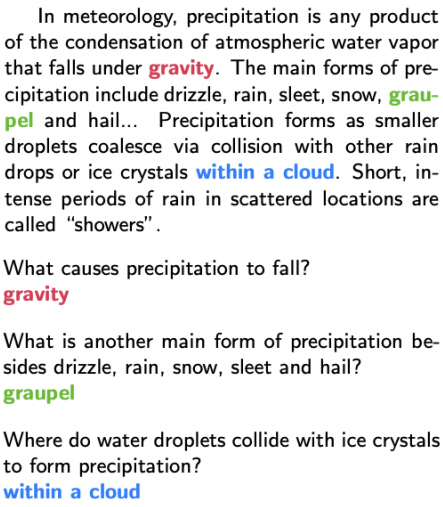


Figure 4: A sample of the SQuAD Dataset as shown in [13]. Presented here in order to gauge a deeper understanding of what the BERT model is encountering while training.

We now imported the "bert-large-uncased-whole-word-masking-finetuned-squad" from the `Transformers` library to implement the pretrained model on the SQuAD dataset. Since this model is BERT Large, it is heavier in comparison to BERT Base, and has **24 encoder blocks** in its architecture. Furthermore, this models architecture can be described as - **1024 hidden dimensions, 16 attention heads and a total of 336 Million parameters**. As a result, even though we did not train this model, it takes a longer time in generating a response for the questions we ran through it as compared to the BERT Base model we also implemented.

3.4 Workflow for BERT Models

The BERT models work on a system of start scores and end scores. A start and end score is assigned to each token in the dataset. The token with the greatest start score is the beginning of the paraphrasing, and the token with the greatest end score indicates the end of the paraphrasing and thus forms our answer. We will visualise this methodology for the benefit of the reader further in the report.

3.5 BIDAF Model

`AllenNLP` library has a pretrained BIDAF model which we used for our project. After loading the model, one is required to pass the data as a context, and questions which the model then

attempts to answer. We evaluated this model with curated data, as well as the SQuaD v2 validation set.

3.6 Format of the Dataset

The SQuAD dataset consists of 536 articles sampled from Wikipedia from which 23,215 individual paragraphs were extracted. The properties of the dataset are as follows – dates and numbers make up 19.8% of the answers, nouns make up 32.6%, noun-phrases make up 31.8%, and other categories make up the remaining 15.8% [14]. A more detailed example can be found in Figure 4. The dataset was found by the creators to be syntactically diverse, which is an advantage.

3.7 Evaluation Metrics

Most question-answering models use two evaluation metrics: exact match (EM) and F1 score. These metrics are evaluated on each question-answer pair and if multiple correct answers are plausible then the maximum score over all possible answers is calculated. In this subsection we describe evaluation metrics used in this project.

3.7.1 Exact Matching(EM) Score

The first evaluation metric we used was the Exact-Matching Score. The way we set up the evaluation for this metric was as follows - if the characters that the model gives in the predicted answer match **exactly** with that of the true answer, the EM score is a 1, else it is 0. This is a binary metric to understand how well our models perform.

3.7.2 F1 Score

F1 Score is the harmonic mean of the precision and recall and a very common metric used in QA problems. F1 score is particularly insightful since it takes into account both the precision as well as the recall. In our case, we use the F1 score in order to measure how well individual words are being predicted against the True Answer. Therefore, the common tokens in the true answer and predicted answer factor into the F1 score.

Here precision can be considered as the ratio between common words between the predicted and True Answer to the total number of words present in the predicted answer. Recall is the ratio between between common words between the predicted and True Answer to the total number of words present in the True Answer. We can mathematically represent these entities as follows:

$$Precision = \frac{\text{CW}}{\text{PA}}$$

$$Recall = \frac{\text{CW}}{\text{TA}}$$

Where,

CW = Common Words between predicted answer and true answer
PA = Number of words in predicted answer
TA = Number of words in true answer

$$F_1 = \frac{2}{precision^{-1} + recall^{-1}}$$

Therefore,

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

4 Experiments

In this section we will illustrate our experiments on a common passage, and set of questions posed on each of the four models. We will now display in the, EM Score, and F1 Score of each model on this passage.

We created a context passage based on the popular children's book series Harry Potter's Wikipedia page, along with a list of 6 common curated questions for each model in varying degrees of difficulty. We have attached the passage in Table 5.

4.1 EM Score Table

Questions	**Baseline Bert**	**Pretrained Model**	**Finetuned Model**	**BIDAF**
Where do Harry and his friends study?	FALSE	TRUE	FALSE	TRUE
What is the main theme of Harry Potter?	FALSE	TRUE	FALSE	TRUE
Who originally published Harry Potter in United Kingdon?	FALSE	TRUE	FALSE	FALSE
how many novels are there in Harry Potter?	FALSE	TRUE	TRUE	TRUE
Who are Harry Potter's friends?	FALSE	TRUE	FALSE	TRUE
Who is Lord Voldemort?	FALSE	TRUE	FALSE	TRUE
Total Count	**0**	**6**	**1**	**5**
Average EM Score	**0**	**1**	**0.16**	**0.83**

Table 1: EM Score on a Harry Potter Context

4.2 F1 Score Table

Questions	**Baseline Bert**	**Pretrained Model**	**Finetuned Model**	**BIDAF**
Where do Harry and his friends study?	0	1	0	1
What is the main theme of Harry Potter?	0	1	0	1
Who originally published Harry Potter in United Kingdon?	0	1	0	0
how many novels are there in Harry Potter?	0,25	1	1	1
Who are Harry Potter's friends?	0,12	1	0	1
Who is Lord Voldemort?	0	1	0	1
Mean F1 Score	**0.06**	**1**	**0.16**	**0.83**

Table 2: F1 Score on a Harry Potter Context

4.3 Interesting Answers

Tabe 3 records some interesting answers given by the models on the set of Harry Potter questions. It can be observed that the baseline model is not able to find an answer in either case; the Finetuned and BIDAF models are able to answer questions albiet often incorrectly; The Pretrained BERT model on the SQuAD dataset however is accurately able to answer questions in both cases. A comprehensive table of the exact answers given by all the models is recorded in Table 6.

Questions	**True Answers**	**Baseline Bert**	**Pretrained**	**Finetuned**	**BIDAF**
Where do Harry and his friends study?	Hogwarts School of Witchcraft and Wizardry	Not Found	Hogwarts School of Witchcraft and Wizardry	harry potter	Hogwarts School of Witchcraft and Wizardry
Who originally published Harry Potter in United Kingdon?	Bloomsbury	Not Found	Bloomsbury	harry potter	Ron Weasley

Table 3: Some interesting answers captured by the models for the Harry Potter questions.

4.4 Visualisations

In this section we will demonstrate the start and end scoring methodology implemented in the BERT models in order to correctly answer questions from the context. We are adding examples from our best performing model, ie, the pretrained model on SQuAD.

Figure 10 records the start and end tokens when the pretrained model is asked the question "Where do Harry and his friends study?". As demonstrated in the image, the token with the highest start score is the first token, and the token with the highest end score is the final token of the answer. This is also demonstrated in Fig 6, which records the pretrained model's answer to the question "What is the main theme of Harry Potter?". Here it can be seen, that the highest start and end scores fall on the same token, which is why the final answer is a single word.

We have also attached similar visualisations for the pretrained for all questions in the B.3 section in the appendix.

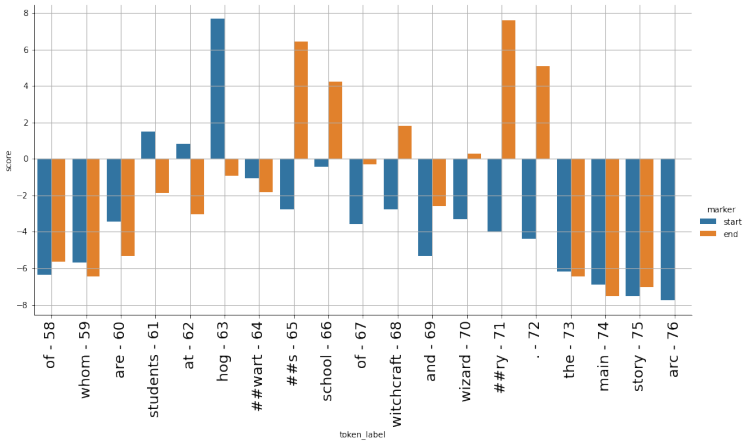


Figure 5: Start and end scores of the tokens of the pretrained model. Answers the question,"Where do Harry and his friends study?"

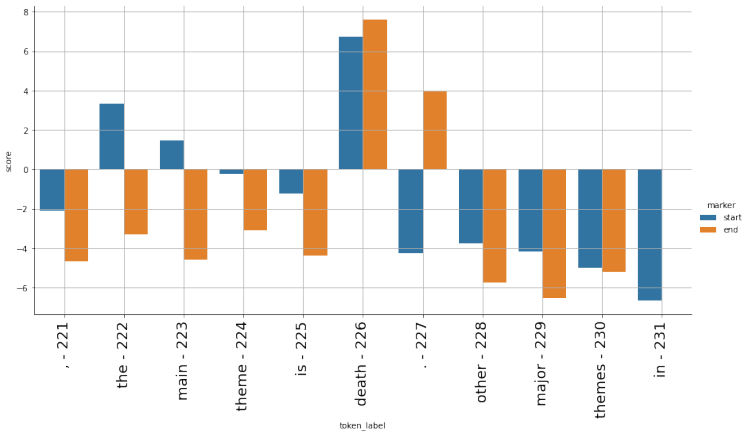


Figure 6: Start and end scores of the tokens of the pretrained model. Answers the question, "What is the main theme of Harry Potter?"

5 Results

In this section we will record the final results achieved after running the 4 models.

Table 4 records the EM and F1 score on the SQUAD 2.0 Validation set for all four models implemented in this project. As expected, the two pretrained models outperform the baseline model as well as the model we trained by tuning the hyperparameters. We can clearly see that the pretrained BERT model outperforms BIDAF model, which outperforms the finetuned model on SQuAD and baseline BERT respectively.

Model Type	**Exact Match** (%)	**Mean F1 Score**
Baseline	0.00	0.025
Finetuned	10.89	0.170
BIDAF	53.72	0.684
Pretrained	63.11	0.767

Table 4: EM Score and Mean F1 for the 4 models we have used in this project. As expected, the pretrained models outperform our baseline and finetuned model

6 Discussion

After creating the four models demonstrated and validating them on the SQuAD v2 dataset, the pretrained BERT model was able to easily outperform the other three models in the as expected, both in metrics of mean F1 score, and EM Score.

One thing that was a great bottleneck in this project was training the BERT model on SQuAD. Each epoch took approximately 3 hours, and a large amount of memory. Training was primarily done on Google Colab which unfortunately crash several times during the training of our model. Since our finetuned model performs better than the baseline, but is unable to outperform the pretrained model – we believe that further training our finetuned model would improve performance greatly. Our batch size was 8 for training the model, and we found it impossible to train the model with a greater batch size due to RAM memory constraints in Google colab. We believe this is a great constraint considering the amount of data in the SQuAD dataset. A greater batch size would likely result in a smoother learning curve and therefore better performance for our finetuned model.
The most intuitive expansion of our project is to train the finetuned model on superior computing capbilities in order to improve performance. We would also like to curate a dataset to fine-tune the baseline BERT model on a specific topic (for example, COVID-19, or Financial Technology) in order to create a more specialised model.

7 Division of labor

In this section, we describe the division of labor during our project. All members contributed more or less equally to the project and its completion. All members of the team were present in meetings. Next, we describe more in detail the individual work of each group member.

Hans Tiwari focused on implementing BERT and worked along with Preetha on the fine-tuned BERT model. In addition, Hans also contributed towards writing this report.

Preetha Datta worked on the BIDAF model. Preetha also worked along with Hans on fine-tuning the BERT model. Finally, Preetha was primarily responsible for writing this report as well.

Since the team had two members, both were largely involved in one another's work and

have had a fairly equal division of labour throughout the course of this project.

8 Acknowledgements

We want to acknowledge and thank the TAs Aku Rouhe and Anssi Moisio for their time and guidance on our project. We would also like to thank our friends Pranav and Yuvraj for their insights into our project and their companionship in pulling off the many all nighters it took to finish this project!

References

[1] M. He, Y. Song, K. Xu, and D. Yu, "On the role of conceptualization in commonsense knowledge graph construction," *arXiv preprint arXiv:2003.03239*, 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

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

[4] J. A. Alzubi, R. Jain, A. Singh, P. Parwekar, and M. Gupta, "Cobert: Covid-19 question answering system using bert," *Arabian Journal for Science and Engineering*, pp. 1–11, 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] H. Face, "Bert-base-uncased · hugging face." [Online]. Available: https://huggingface.co/bert-base-uncased

[7] R. Horev, "Bert explained: State of the art language model for nlp," Nov 2018. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[8] S. Persson, "Paper summary-bert: Pre-training of deep bidirectional transformers for language understanding," May 2021.

[9] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, "End-to-end open-domain question answering with bertserini," *arXiv preprint arXiv:1902.01718*, 2019.

[10] C. Qu, L. Yang, M. Qiu, W. B. Croft, Y. Zhang, and M. Iyyer, "Bert with history answer embedding for conversational question answering," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 1133–1136.

[11] F. Alloatti, L. Di Caro, and G. Sportelli, "Real life application of a question answering system using bert language model," in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 2019, pp. 250–253.

[12] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[13] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[14] J. Wei, "The quick guide to squad," Oct 2020. [Online]. Available: https://towardsdatascience.com/the-quick-guide-to-squad-cae08047ebee

[15] H. Mozannar, K. E. Hajal, E. Maamary, and H. Hajj, "Neural arabic question answering," *arXiv preprint arXiv:1906.05394*, 2019.

[16] T. Möller, A. Reina, R. Jayakumar, and M. Pietsch, "Covid-qa: a question answering dataset for covid-19," 2020.

[17] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

A Code

All the code for our project can be found on this github repository

B Experimental Task

B.1 Context Paragraph

The context paragraph is recorded in the following table.

Harry Potter is a series of seven fantasy novels written by British author J. K. Rowling. The novels chronicle the lives of a young wizard, Harry Potter, and his friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of Witchcraft and Wizardry. The main story arc concerns Harry's struggle against Lord Voldemort, a dark wizard who intends to become immortal, overthrow the wizard governing body known as the Ministry of Magic and subjugate all wizards and Muggles (non-magical people). The series was originally published in English by Bloomsbury in the United Kingdom and Scholastic Press in the United States. All versions around the world are printed by Grafica Veneta in Italy. A series of many genres, including fantasy, drama, coming of age, and the British school story (which includes elements of mystery, thriller, adventure, horror, and romance), the world of Harry Potter explores numerous themes and includes many cultural meanings and references. According to Rowling, the main theme is death. Other major themes in the series include prejudice, corruption, and madness.

Table 5: Context Passage. This paragraph has been curated from popular children's book series Harry Potter's Wikipedia page.

B.2 Answers

Table 6: All answers given by the four models on the set of questions based on the Harry Potter context

Questions	**True Answers**	**Baseline Bert**	**Pretrained**	**Finetuned**	**BiDAF**
Where do Harry and his friends study?	Hogwarts School of Witchcraft and Wizardry	Not Found	Hogwarts School of Witchcraft and Wizardry	harry potter	Hogwarts School of Witchcraft and Wizardry

What is the main theme of Harry Potter?	death	magical people) . the series was originally published in english by bloomsbury in the united kingdom...including fantasy , drama	death	harry potter	death
Who originally published Harry Potter in United Kingdon?	Bloomsbury	Not Found	Bloomsbury	harry potter	Ron Weasley
how many novels are there in Harry Potter?	Seven	series of seven fantasy novels written by	Seven	seven	seven
Who are Harry Potter's friends?	Hermione Granger and Ron Weasley	a young wizard , harry potter , and...press in the united states . all versions	Hermione Granger and Ron Weasley	-	Hermione Granger and Ron Weasley
Who is Lord Voldemort?	a dark wizard	Not Found	a dark wizard	harry potter	a dark wizard

Table 6: All answers given by the models on the questions posed on the Harry Potter context

B.3 Visualisations

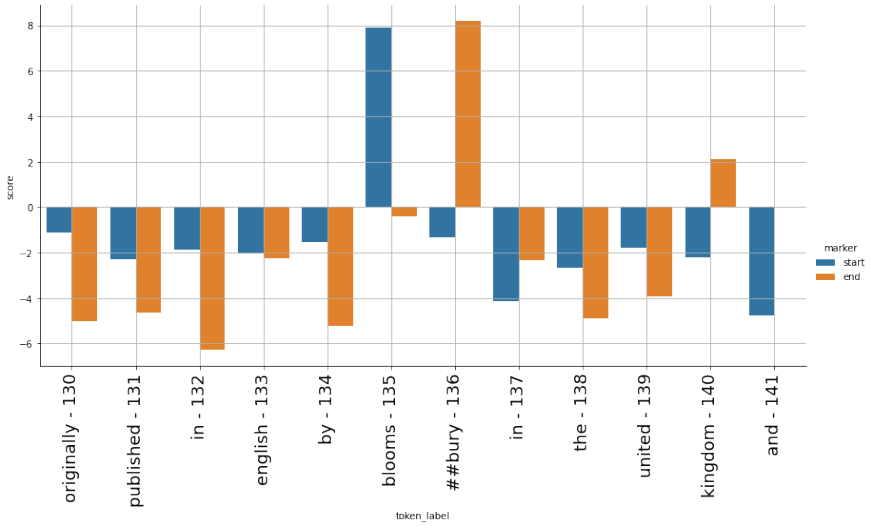


Figure 7: Start and end scores of the tokens of the pretrained model. Answers the question,"Who originally published Harry Potter in United Kingdon?"

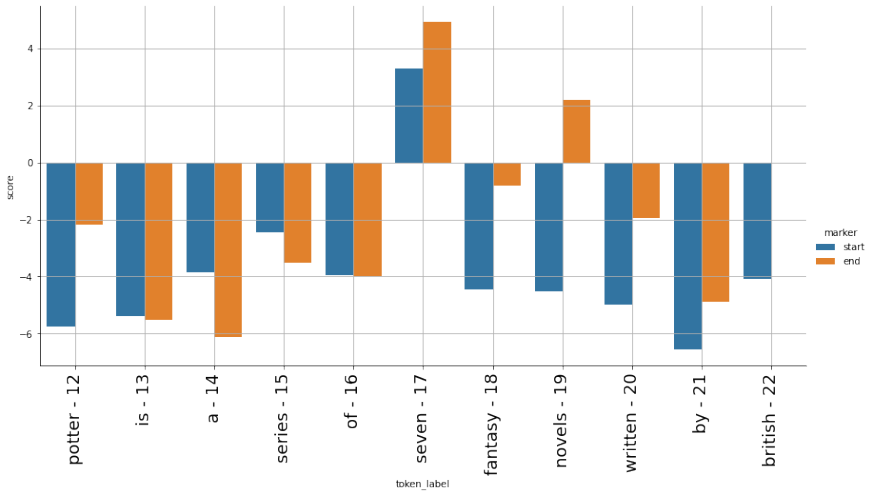


Figure 8: Start and end scores of the tokens of the pretrained model. Answers the question,"how many novels are there in Harry Potter?"

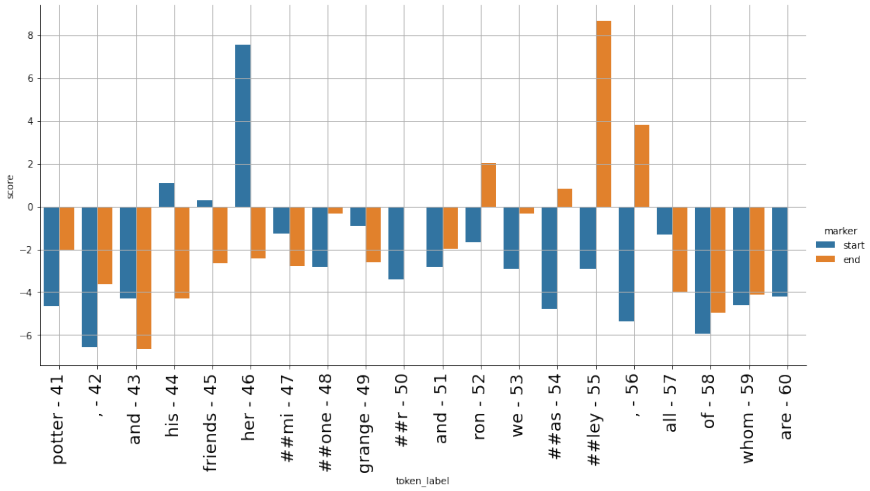


Figure 9: Start and end scores of the tokens of the pretrained model. Answers the question,"Who are Harry Potter's friends?"

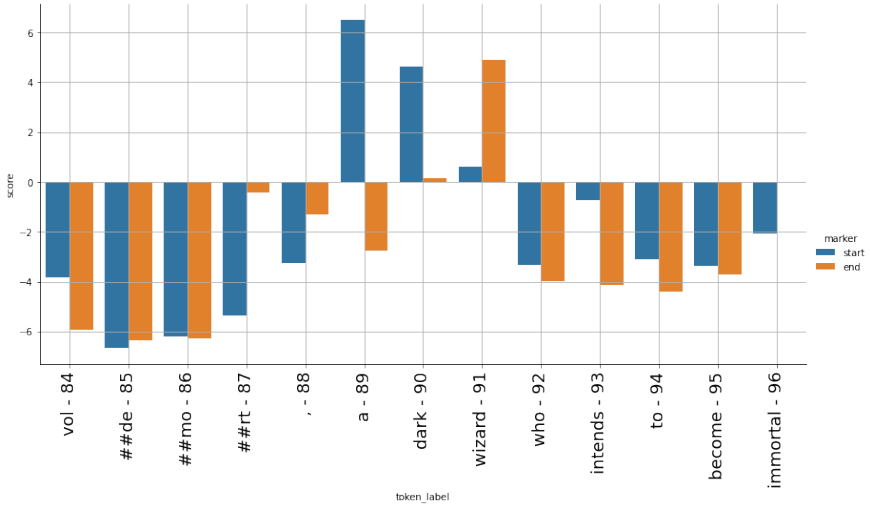


Figure 10: Start and end scores of the tokens of the pretrained model. Answers the question,"Who is Lord Voldemort?"