Final Report

Enhancing Question-Answering Systems with BERT

Project Group 50

Ayushi Chakrabarty, Cameron Potter, Kshitij Pathania, Prateek Yadav, Sneha Maheshwari

{achakrabarty8, cpotter8, kpathania3, p34, smaheshwari63}@gatech.edu

Introduction

In the evolving landscape of Natural Language Processing (NLP), the quest for models that can understand and generate human-like responses has become paramount. Among the myriad of models that have emerged, BERT [4] (Bidirectional Encoder Representations from Transformers) stands out as a revolutionary architecture that has set new benchmarks in a range of NLP tasks. The underlying Transformer architecture, introduced by Vaswani et al. [15], forms the basis of BERT, bringing a significant shift in how deep learning models process sequences. Specifically, in the domain of Question-Answering (QA) systems, the adaptability and prowess of BERT can lead to significant enhancements in accuracy, contextual understanding, and response generation. This paper delves deep into the nuances of integrating BERT into QA tasks, showcasing its potential to reshape the way machines understand and answer questions. As we navigate the landscape of deep learning, this integration aligns with the evolutionary trajectory outlined by Goodfellow, Bengio, and Courville [5], marking a transformative step towards endowing machines with more human-like conversational abilities.

Background

In the recent years, the development of advanced models and methodologies for natural language processing has been a focal point in various research domains. An influential [15] paper by Vaswani et al. introduces the Transformer architecture, a groundbreaking network structure relying solely on attention mechanisms showcasing superior performance in translation tasks. Simultaneously, the rise in demand for round-the-clock online services, notably during the pandemic, led to the emergence of innovative approaches like BIRD-QA [1]. This domain-specific question answering framework departs from rule-based strategies, leveraging BERT-style pre-trained models and domain-specific knowledge bases to achieve significant accuracy in reading comprehension tasks. Addressing the intricacies of commonsense reasoning in reading comprehension, [7] introduces COSMOS QA, a large-scale dataset designed for this purpose. Unlike traditional datasets, it emphasizes questions that necessitate interpreting implicit causes and effects in diverse everyday narratives, exposing a notable performance gap between machines and humans. The QASC study [8] introduces a novel multi-hop reasoning dataset, focusing on the retrieval and composition of facts from

a large corpus. Implementing a two-step approach to enhance retrieval, the study showcases improvements in state-of-the-art language models. However, persistent gaps in reasoning and retrieval compared to human performance underscore the challenges in this domain.

Exploring the domain of unsupervised question answering, recent research unveils innovative methodologies such as semi-supervised data augmentation, transfer learning, and clustering techniques. The unsupervised approach introduced in [10] creatively generates context, question, and answer triplets, achieving commendable performance on the SQuAD dataset without relying on labeled data. Additionally, an unsupervised question generation method is proposed in [12]. Utilizing freely available news summary data, this method generates diverse questions, contributing to the training of a high-performing QA model.In the domain of short text clustering, [3] introduces a deep embedded approach utilizing an auto-encoder for sentence distributed embedding, outperforming existing correlation clustering methods. Furthermore, [6] addresses the need for systematic analysis of attention patterns in multi-head attention mechanisms. Introducing a feasible unsupervised clustering method, the study provides valuable insights into attention interpretability. [16] innovatively tackles extreme long-range dependencies in sequences with a clustering-based sparse Transformer, achieving state-of-the-art results in question answering benchmarks. Transitioning to transfer learning models, the study conducted by Li et al. [11] addresses the limitations of extractive MRC, introducing effective methods for fine-tuning with commonsense representations. Additionally, [13] explores the impact of intermediate-task training on pre-trained models, emphasizing high-level reasoning skills. [2] introduces a framework for enhancing QA retrieval in domain-specific search engines, demonstrating improved F1 score and accuracy. Finally, the paper on [9] extends zero-shot transfer learning to multiple-source settings, mitigating knowledge loss and improving performance on commonsense reasoning benchmarks.

Therefore, these studies collectively contribute to advancing the understanding and capabilities of supervised and unsupervised methods in various natural language processing applications utilizing BERT.

Problem Definition

The primary challenge is to investigate and evaluate how BERT can be effectively integrated into existing or novel QA frameworks to enhance their performance. This includes:

1. How can BERT's bidirectional understanding of context be harnessed to improve the accuracy of QA systems?

2. What modifications or fine-tuning techniques are required to adapt BERT specifically for diverse QA tasks, considering the variability in question types and domains?

3. How can the scalability and efficiency of QA systems be maintained or improved upon integrating the computationally intensive BERT model?

4. In what ways can BERT's capabilities be leveraged to make QA systems more robust against ambiguous, misleading, or poorly framed questions?

Data Collection

The data for this project was obtained from the Stanford Question Answering Dataset [14] (SQuAD 2.0). This dataset includes a set of questions and answers based on a series of Wikipedia articles. We used two subsets of this dataset: the training set (train-v2.0.json) and the development set (dev-v2.0.json). These files were downloaded and loaded into our environment for processing and training our model. Preprocessing steps included tokenizing contexts and questions and mapping answer start and end positions in the token space. In addition to preprocessing, we analyzed the distribution of the lengths of contexts and questions to gain insight into the dataset. The length of a context and its corresponding questions can significantly impact the complexity of the QA task.

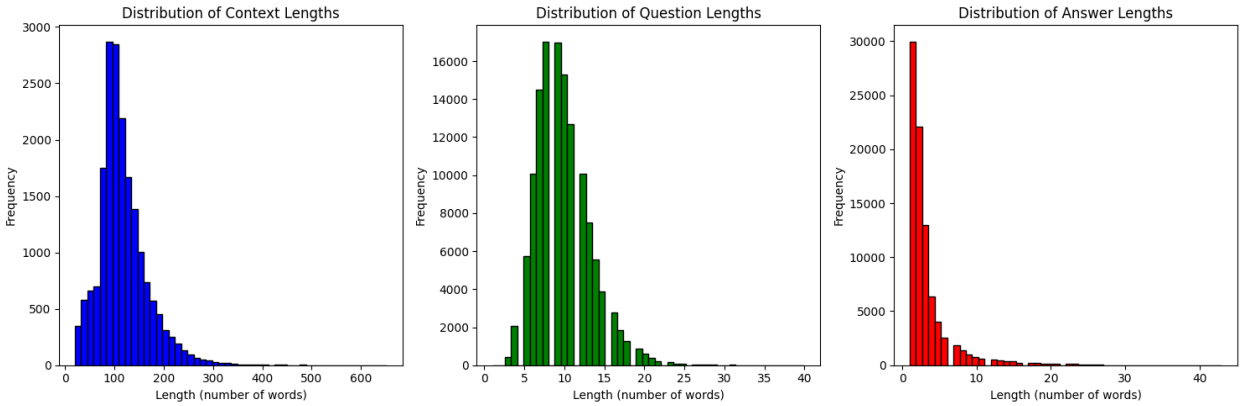

Figure 1: Distribution of context, question and answer lengths in the SQuAD 2.0 training set.

Methods

Our methodology involved a series of steps from data loading and preprocessing to model training and fine-tuning. The following details each step:

Data Loading and Preprocessing

We loaded our data from the SQuAD v2.0 dataset, which includes a training set (`train-v2.0.json`) and a development set (`dev-v2.0.json`). The dataset consists of contexts, questions, and their corresponding answers. The preprocessing steps included:

- Parsing each context, question, and answer from the dataset.
- Using the `AutoTokenizer` from the Hugging Face Transformers library to tokenize the contexts and questions. We chose the "distilbert-base-uncased" model for tokenization.
- Encoding the answers by mapping their character start and end positions to token positions.
- Handling cases where the answer passages were truncated due to tokenization limits.

Dataset Preparation

A custom dataset class was implemented to manage the tokenized encodings. This class provided mechanisms to retrieve individual data points and their lengths, crucial for creating batches during training and evaluation.

Model and Training

For the question-answering task, we used the `AutoModelForQuestionAnswering` from the Hugging Face library, again utilizing the "distilbert-base-uncased" model. We did the manual tuning for identifying optimal hyperparameters. The training process was carried out with the following hyperparameter combinations:

- Hyperparameters: Number of Epochs (`N_EPOCHS`) = (5, 10, 20), (`Learning_Rate`) = (1e-5, 5e-5), (`Weight_Decay`) = 0.01, (`Batch_Size`) = (16, 32).
- The optimizer used was AdamW, adhering to the specified learning rate and weight decay.
- Training involved iterating over batches of the training dataset, performing forward and backward passes, and updating the model parameters.
- We tracked the loss at each step, which can be visualized to understand the model's learning progress over epochs.

The given line plot depicts the training loss across five different combinations of hyperparameters i.e., (5, 5e-5, 16), (5, 5e-5, 32), (10, 1e-5, 16), (10, 1e-5, 32), and (20, 1e-5, 16) with the parameters defined in the order of (epochs, learning rate, batch size) respectively.

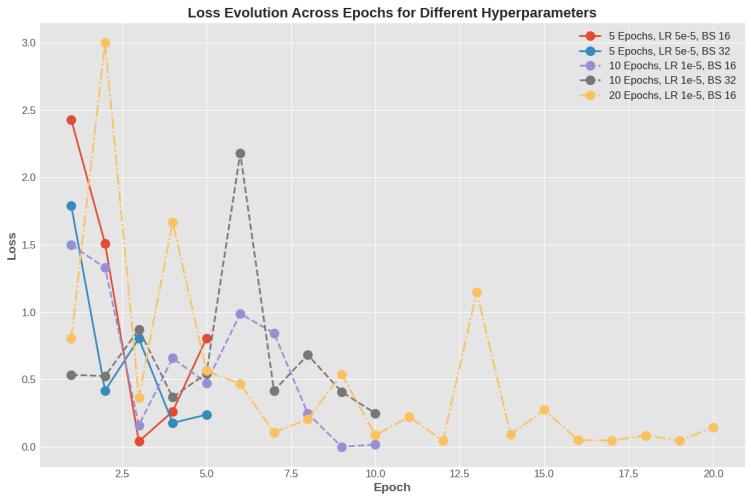

Figure 2: Loss Evolution Across Epochs for Different Hyperparameters

On observing Figure 2 we find that the nature of the loss vs. epoch plot is stochastic in nature across the five different combinations. This can be attributed to the inherent randomness introduced by variability in the training process of neural network algorithms. The plot shows how the loss decreases over epochs highlighting performance differences based on hyperparameter choices. For example, in the case of (5, 5e-5, 32), we observe that from Epoch 1 to Epoch 5, the loss drops from 1.79 to 0.239. Similarly, for (10, 1e-5, 16) the loss drops from 1.5 to 0.0168 and for (20, 1e-5, 16) the loss drops from 0.808 to 0.146. Therefore, for each of these combinations, the loss significantly decreases from the first to the final epoch providing a strong indication that the model is learning effectively and improving in its ability to make accurate predictions. Notably, the 20-epoch experiment with a learning rate of 1e-5 and batch size 16 exhibits fluctuations but generally achieves lower training losses over time, suggesting potential sensitivity to hyperparameter settings. As lower training loss is often an indicator of better model convergence and performance during training, we consider (20, 1e-5, 16) to give us the best performance.

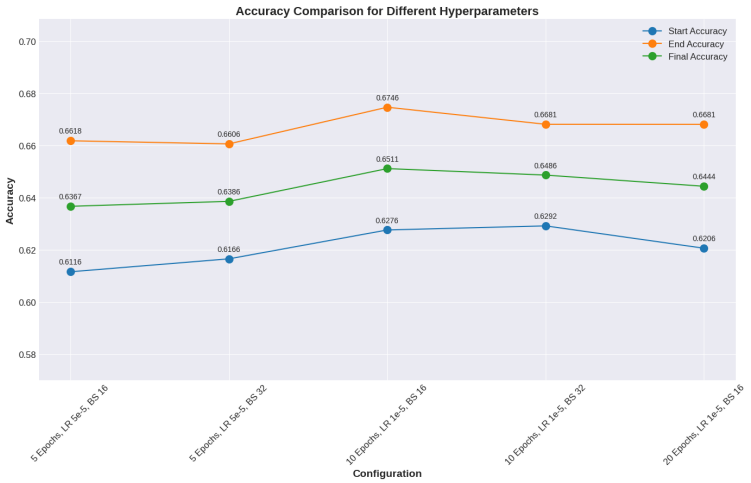

Figure 3: Accuracy Comparison for Different Hyperparameters

Figure 3 illustrates the final accuracies achieved by the model configurations under investigation. It is evident from the plot that the hyperparameter combination of 20 epochs, a learning rate of 1e-5, and a batch size of 16 leads to the highest start, end, and overall accuracy, corroborating the findings suggested by the loss trends observed in Figure 2.

Results and Discussion

The evaluation of our fine-tuned BERT model on the test dataset provided insightful metrics into its performance. We observed substantial accuracy in predicting the start and end positions of answers, which are critical for the efficacy of Question-Answering systems. The F1 score was computed across batches, offering a comprehensive measure that captures both the precision and recall of the model's predictions.

In addition to accuracy metrics, we monitored the latency of the model's responses, which is pivotal for real-time applications. Initially, a significant outlier in the latency measurements

was identified, which skewed the overall representation of the model's responsiveness. After adjusting for these outliers, we obtained a more representative overview of the model's average response time.

Comparative Analysis

In this section, we present the results of our experiments, comparing the performance of our fine-tuned model on SQuAD with two baseline models not fine tuned over the SQuAD dataset: `distilbert-base-uncased` and `distilbert-base-uncased-distilled-squad`.

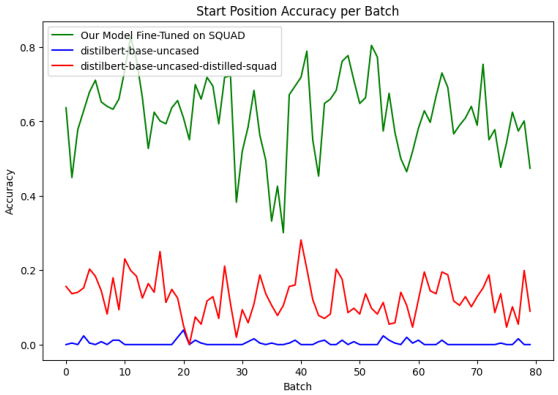

Figure 4: Start Position Accuracy per Batch

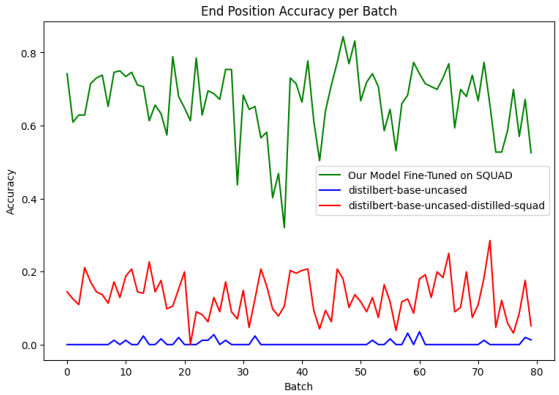

Figure 5: End Position Accuracy per Batch

Figure 4 shows the start position accuracy per batch for each model. Our fine-tuned model consistently outperforms the baseline models, achieving higher accuracy throughout the evaluation.

Similarly, Figure 5 displays the end position accuracy per batch. Our model maintains superior accuracy compared to the baseline models during the evaluation process.

The comparative analysis of the models presents a clear contrast in performance. Both 'distilbert-base-uncased' and 'distilbert-base-uncased-distilled-squad' exhibited poor results on the question-answering task. This underperformance is largely due to the absence of fine-tuning on the SQuAD dataset. The SQuAD dataset is instrumental for training models to comprehend and respond to questions with contextual accuracy. Since the baseline models were not fine-tuned on this dataset, they lack the refined capabilities essential for the detailed understanding and processing of natural language queries.

Conversely, our model, which is a fine-tuned version of 'distilbert-base-uncased', demonstrated significantly better performance. The fine-tuning process involves adjusting the model's parameters specifically to the types of patterns and tasks found in the SQuAD dataset. This enables the model to develop a more acute understanding of the intricacies involved in question answering, such as the relevance of context and the nuances of linguistic structure. As a result, the fine-tuned model was able to outperform the baseline models, showcasing the effectiveness of fine-tuning on specialized datasets for tasks that demand a deep understanding of language and context.

The F1 score per batch is illustrated in Figure 6. Our fine-tuned model consistently

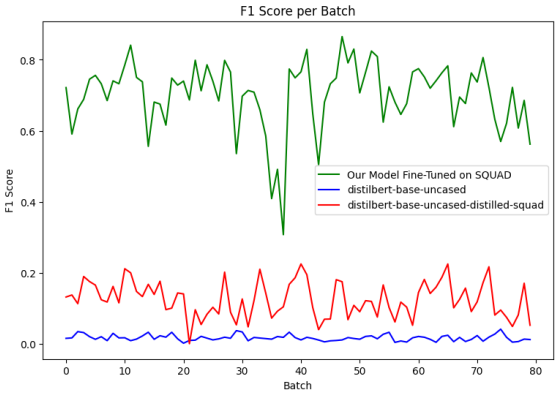

Figure 6: F1 Score per Batch

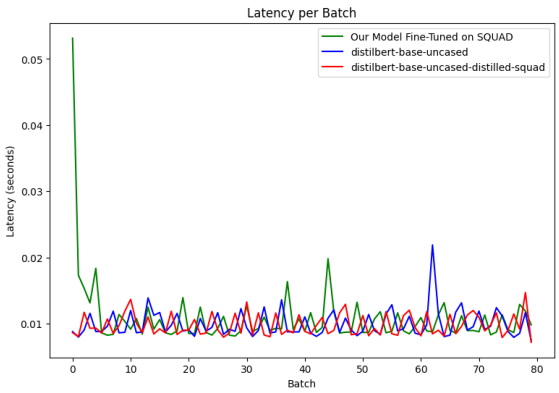

Figure 7: Latency per Batch

achieves higher F1 scores compared to the baseline models, indicating better overall performance.

Figure 7 presents the latency per batch for each model. While our model demonstrates slightly higher latency, the trade-off in performance justifies this difference.
Adding to the above, we have summarized our cumulative results as follows:

- **Start Accuracy:** Our model achieved a start accuracy of 0.6194, indicating its effectiveness in identifying the beginning of the answer span.
- **End Accuracy:** The end accuracy was 0.66687, showing the model's ability to correctly pinpoint the end of the answer span.
- **F1 Score:** We achieved an F1 score of 0.7009, reflecting a strong balance between precision and recall in our predictions.
- **Average Latency:** After adjusting for outliers, the average latency of the model's responses was approximately 0.05 seconds, demonstrating its suitability for real-time applications.

Robustness Evaluation on Ambiguous Queries

To assess the robustness of our model, we conducted a comprehensive evaluation on a set of ambiguous queries. These queries were designed to test the model's ability to handle various forms of ambiguity, including multiple valid answers, subjective interpretations, and cases where the correct answer might depend on contextual nuances. The objective was to gauge the model's performance in scenarios where a single, clear-cut answer might not exist.

Test Setup: We formulated a set of ambiguous queries covering diverse topics and paired them with corresponding contexts. The model was then evaluated on its ability to provide relevant and accurate answers to these queries. The evaluation was performed using the provided code snippet that leverages the model's predictions and compares them against expected answers.

Ambiguous Queries and Contexts: Here are the ambiguous queries and their respective contexts used in the evaluation:

1. **Query:** Who led the expedition?
   Context: In 1804, Meriwether Lewis and William Clark led an expedition to explore the newly acquired western portion of the United States after the Louisiana Purchase. Meanwhile, in the same year, Napoleon Bonaparte declared himself the emperor of France.

2. **Query:** What is the capital of Australia?
   Context: Sydney is the largest city in Australia and is often mistaken for the capital. However, the actual capital of Australia is Canberra.

3. **Query:** When was the treaty signed?
   Context: The treaty, which ended the long-standing war, was signed after prolonged negotiations. The war had lasted for over a decade, with significant losses on both sides.

4. **Query:** What causes the disease?
   Context: The disease, known as Parkinson's, is often confused with Alzheimer's due to similar symptoms. However, Parkinson's is primarily caused by the loss of dopamine-producing cells in the brain.

5. **Query:** Who wrote Hamlet?
   Context: Hamlet, a famous play written by William Shakespeare, has been adapted into various films and books. Some of these adaptations were done by directors like Laurence Olivier and Kenneth Branagh.

6. **Query:** What is the tallest mountain?
   Context: Mount Everest is often cited as the tallest mountain in the world. However, Mauna Kea in Hawaii is technically the tallest when measured from its base underwater.

7. **Query:** What is the heart of the city?
   Context: The city is known for its vibrant central district, often referred to as the heart of the city, which includes both the historical town square and the main business district.

To quantify the model's performance, we utilized standard natural language processing evaluation metrics, including accuracy, precision, recall, and F1 score. These metrics were calculated after filtering out instances where the expected answer was not applicable or subjective.

The results of the robustness evaluation are visualized in Figure 8. The plot illustrates the performance of the model on different queries, sorted by their corresponding scores. Each query is annotated with its performance score and a snippet of the context in which it was evaluated.

Discussion: The evaluation revealed the model's strengths and weaknesses in handling ambiguous queries. Some queries, such as those with clear-cut answers or well-defined criteria, achieved high scores. However, the model faced challenges in cases where answers were subjective or dependent on nuanced contextual understanding.

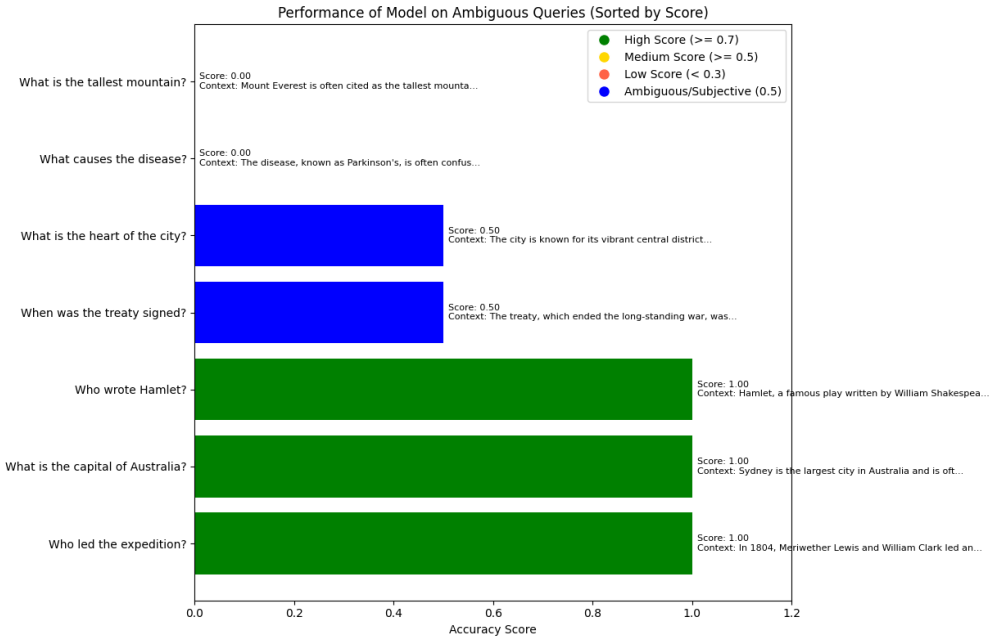

Figure 8: Performance of the Model on Ambiguous Queries (Sorted by Score)

This analysis contributes valuable insights into the model's robustness and highlights areas for potential improvement. Understanding the model's limitations in handling ambiguity is crucial for its practical deployment in real-world applications. The robustness evaluation on ambiguous queries serves as a critical component of assessing the overall reliability of our model. By systematically exploring its performance on diverse and challenging queries, we gain a deeper understanding of its capabilities and limitations.

Conclusion

In the vast landscape of Natural Language Processing (NLP), BERT emerges as a groundbreaking model, particularly shining in the domain of Question-Answering (QA) systems. Its remarkable adaptability holds the promise of significantly enhancing accuracy and contextual understanding. This study delves into the seamless integration of BERT into QA tasks thereby unveiling its potential to revolutionize how machines comprehend and respond to a diverse array of questions. The exploration includes a series of experiments, where the proposed fine-tuned model on SQuAD surpasses two baseline models not fine-tuned on the same dataset. This fine-tuned model exhibits superior performance by boasting start accuracy at 0.6194, end accuracy at 0.66687, an impressive F1 score of 0.7009, all achieved with a latency of just 0.05 seconds. Moreover, the model evaluation provides nuanced insights into its strengths and challenges. While performing well in addressing clear questions, the model encounters difficulties with subjective or context-dependent queries. Recognizing and understanding these limitations becomes pivotal for practical applications thereby enabling researchers to take informed decisions for the continual enhancement of NLP.

Timeline

The project's timeline was strategically divided into three distinct phases, spanning across the entire semester. This structured approach was designed to efficiently harness the capabilities of BERT for enhancing Question-Answering tasks within the allocated time frame. Below is a link to our detailed Gantt chart, which provides a visual representation of the project's timeline and milestones: Gantt Chart

Contribution table

Student Name	**Contributed Aspects**
Ayushi Chakrabarty	Problem Definition, Documentation, Report Writing
Cameron George Potter	Ideation of the project theme, Methods, Model Training
Kshitij Pathania	Model evaluation, Results Compilation, Data visualisation
Prateek	Problem Definition, Ideation of the project theme, Documentation
Sneha Maheshwari	Problem Definition, Report Writing, Documentation

Table 1: Contributions of team members

References

[1] Chen, Y., and Zulkernine, F. Bird-qa: a bert-based information retrieval approach to domain specific question answering. In *2021 IEEE International Conference On Big Data (Big Data)* (2021), IEEE, pp. 3503–3510.

[2] Chopra, A., Agrawal, S., and Ghosh, S. Applying transfer learning for improving domain-specific search experience using query to question similarity. In *Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence* (2020), pp. 1–8.

[3] Dai, Z., Li, K., Li, H., and Li, X. An unsupervised learning short text clustering method. In *Journal of Physics: Conference Series* (2020), vol. 1650, IOP Publishing, p. 032090.

[4] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding.

[5] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[6] Guan, Y., Leng, J., Li, C., Chen, Q., and Guo, M. How far does bert look at: Distance-based clustering and analysis of bert ′ s attention. *arXiv preprint arXiv:2011.00943* (2020).

[7] Huang, L., Bras, R. L., Bhagavatula, C., and Choi, Y. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277* (2019).

[8] Khot, T., Clark, P., Guerquin, M., Jansen, P., and Sabharwal, A. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 8082–8090.

[9] Kim, Y. J., Kwak, B.-w., Kim, Y., Amplayo, R. K., Hwang, S.-w., and Yeo, J. Modularized transfer learning with multiple knowledge graphs for zero-shot commonsense reasoning. *arXiv preprint arXiv:2206.03715* (2022).

[10] Lewis, P., Denoyer, L., and Riedel, S. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980* (2019).

[11] Li, R., Wang, L., Jiang, Z., Liu, D., Zhao, M., and Lu, X. Incremental bert with commonsense representations for multi-choice reading comprehension. *Multimedia Tools and Applications 80* (2021), 32311–32333.

[12] Lyu, C., Shang, L., Graham, Y., Foster, J., Jiang, X., and Liu, Q. Improving unsupervised question answering via summarization-informed question generation. *arXiv preprint arXiv:2109.07954* (2021).

[13] Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628* (2020).

[14] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text, 2016.

[15] Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017), Curran Associates Inc.

[16] Wang, S., Zhou, L., Gan, Z., Chen, Y.-C., Fang, Y., Sun, S., Cheng, Y., and Liu, J. Cluster-former: Clustering-based sparse transformer for long-range dependency encoding. *arXiv preprint arXiv:2009.06097* (2020).