Medical Question Answer Classification using MedQA Dataset from Hugging Face

Sai Spandana Sannapureddy and Prashanth Chitturi

Abstract

This report details an investigation into applying Transformer-based models for Medical Question Answering (MedQA), concentrating especially on the multiple-choice format that is common in medical exams. The project initially focused on using domain-specific transformer models, BioBERT and BioGPT, which were pre-trained on biomedical text. We experimented with different fine-tuning strategies, such as fully fine-tuning the models, adding a custom classification head, and freezing or partially freezing the transformer layers. Although we tried a few hyperparameter settings, full hyperparameter optimization was not done due to limited time and resources. Despite these efforts, the models consistently performed poorly, with accuracy stuck around 30%. Following this, we shifted the project from a classification task to a generative task, since the target depended on the correct answer's position rather than direct classification. We fine-tuned BioGPT with its original language modeling (LM) head for this generative setting. Subsequently, we attempted prompting techniques (zero-shot, few-shot, Chain-of-Thought, and log-likelihood scoring) on BioGPT, and later BioBART, but these approaches also failed to produce good results. As these models lacked sufficient reasoning capacity, the project shifted to using Large Language Models (LLMs), namely GPT-3.5-turbo, GPT-4o-mini, and Gemini-2.0-flash-lite. Using identical prompting strategies that previously failed on smaller models, these LLMs achieved significantly higher accuracy, exceeding 75%. This outcome strongly suggests that the complex reasoning and nuanced understanding required for the MedQA multiple-choice task are capabilities primarily present in large-scale models. Despite their domain-specific pre-training, the smaller, domain-specific models likely failed because of architectural limitations, insufficient parameter scale, restrictive context windows, and challenges in adapting them for complex comparative reasoning.

Introduction

The goal of this project was to use Natural Language Processing (NLP) to help a model answer complex medical multiple-choice questions (MCQs) by understanding the question, thinking through the options, and choosing the correct answer similar to how it's done in real medical exams.

At first, we treated the task as a classification problem, using domain-specific transformer models like BioBERT[1][1] and BioGPT[2][2]. These models were pre-trained on biomedical text, so we expected them to perform well. BioBERT, built on the encoder-based BERT[3] model, specializes in understanding text, while BioGPT, as a generative model, was trained for biomedical language generation.

After observing poor performance with classification approaches, we realized that the MedQA task should not be a classification task, where the target depends on the position of the correct answer. So, we shifted our approach to the generative task. Since BioBERT is not a generative model, we continued the generative experiments using BioGPT and BioBART[4], which is an encoder-decoder model. However, even after shifting to a generation-based strategy, these models struggled to achieve good performance.

[1] https://github.com/naver/biobert-pretrained

[2] https://huggingface.co/docs/transformers/en/model_doc/biogpt

Ultimately, we transitioned to using Large Language Models (LLMs) like GPT-3.5-turbo[5, 6], GPT-4o-mini[7], and Gemini-2.0-flash-lite[8]. These LLMs, trained on huge, diverse datasets, with strong abilities in reasoning and following complex instructions, often outperform smaller, specialized models even on tasks in specific fields like medicine. Using the same prompting techniques that had previously failed on smaller models, these LLMs achieved much higher accuracies, exceeding 75%.

Overall, this project compared the performance of specialized biomedical models and general-purpose LLMs on complex medical MCQs. We learned that while domain-specific training helps with vocabulary, deep reasoning and flexible understanding which are strengths of larger LLMs are much more important for success in this task.

This report describes our journey through the experiments, the surprising challenges we faced with the smaller models, and how switching to larger LLMs led to much better performance, along with an analysis of why this shift made such a difference.

Dataset

The primary dataset used in this project was the MedQA dataset, accessed through Hugging Face (VodLM/medqa)[9][3]. The MedQA dataset contains multiple-choice questions from medical licensing examinations and is divided into two subsets -

'us' — questions from United States medical exams.

'tw' — questions from Taiwan medical exams.

For this project, we exclusively used the 'us' subset. The 'us' subset includes:

10.2k of training samples

1.27k of validation samples

1.27k of test samples

Each instance in the dataset includes:

- question - A medical question.
- metamap - A preprocessed version of the question highlighting biomedical entities.
- answers - An array of four multiple-choice options.
- target - Index of the correct answer in "answers" array, which is an integer from 0 to 3.

The MedQA dataset is considered a challenging benchmark as it requires not only domain-specific medical knowledge but also reasoning abilities to correctly answer the multiple-choice questions.

Methodology

The project underwent various stages, with the approach being modified in response to experimental results. Initially, we treated MedQA as a classification problem and fine-tuned domain-specific models with a classification head. After observing poor results, we shifted to a generative approach as the target here depends on the position of the correct answer. Although there was a slight improvement, the generative model produced few invalid outputs. We then tried prompt-based techniques, applying zero-shot and few-shot prompting to domain-specific models. Due to continued poor performance, we moved to Large Language Models (LLMs) with the same prompting strategies, which significantly improved the results.

[3] https://huggingface.co/datasets/VodLM/medqa/viewer/us

Phase 1 - Classification Approach

For BioBERT (Fig: 1), We ran three major fine-tuning experiments and tried to make it better at answering multiple-choice medical questions from the MedQA dataset. In each experiment, we changed how we trained the model, how we set up the inputs, and how we made the model give its final answers.

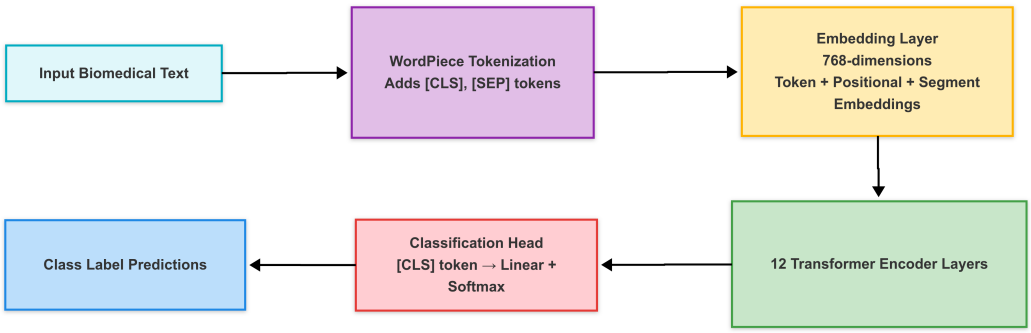

Fig. 1. BIOBERT Architecture

For all three experiments, the first step was to prepare the input. We took each medical question and paired it individually with each of its four answer choices. For each (question, option) pair, we combined them into a single text sequence before giving it to the model. The structure was simple: we started with a [CLS] token, then the question, then a [SEP] token, followed by the answer option, and finally another [SEP] token. We made sure to handle padding properly when the sequence was short and to apply truncation when it was too long. Depending on the experiment, we adjusted the maximum sequence length — setting it to 128, 256, or 384 tokens — so that the model could manage both short and long questions. After tokenization, we reshaped the data so that the model could process each question and all of its options together in a single pass during training and evaluation.

In the first experiment, we fine-tuned the entire BioBERT model using the Hugging face's transformer model AutoModelForMultipleChoice[10] setup. We trained it for 3 epochs with a learning rate of 2e-5, batch sizes of 8 (training) and 16 (evaluation), and applied weight decay of 0.01. The model selected the answer option with the highest score (logit). However, even after full fine-tuning, the model's accuracy stayed around 30% on validation and test sets.

In the second experiment, we froze all BioBERT layers and only trained a new classification head made of two linear layers with a ReLU activation and dropout. We trained the head for 20 epochs with a batch size 16, But the accuracy again stayed near 30%, meaning freezing BioBERT and training only the head was not effective either.

In the third experiment, we partially unfroze BioBERT by allowing only the top two transformer layers to update, while keeping the rest frozen. We also changed the input by adding medical concepts extracted from MetaMap, separated with a [META] token, and used a maximum sequence length of 384 tokens. We trained for 4 epochs with a learning rate of 1e-5 and batch size 16. Even with partial fine-tuning and extra information, still the model's performance stayed close to 30%.

BioGPT (Fig: 2) is originally designed for generative tasks, not classification. To make it work for our multiple-choice classification problem, we loaded the base model using Hugging face's AutoModel[10], which loads the base BioGPT without any language modeling (LM) head. On top of it, we built a

custom classification head made up of a dropout layer, a normalization layer, and a linear layer that maps the models output to four classes.

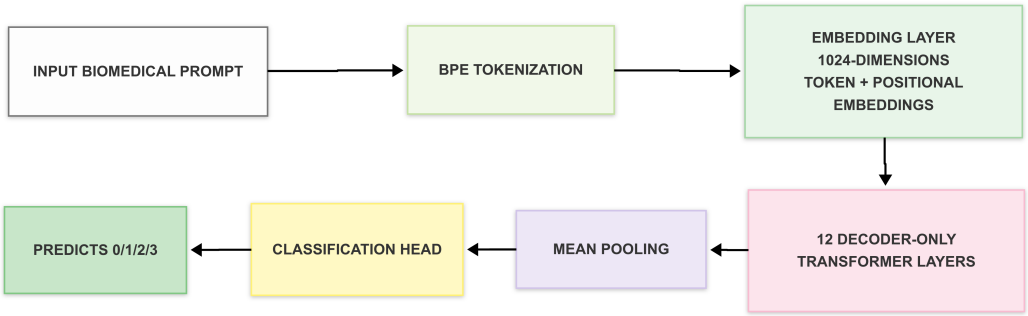

Fig. 2. BIOGPT Architecture for Classification

For inputs, we combined the medical question and the four answer choices into a single text sequence. We then tokenized this sequence using AutoTokenizer[10], setting the tokenizer to apply left padding and experimenting with different maximum sequence lengths, like 256, 384, 512 and 600 tokens, to find the best one. Then the input was passed to embedding layer of the model, where each token will be converted into a 1024-dimensional embeddings. These embeddings are then processed through 12 decoder-only transformer layers[11] of the model to capture the contextual meaning. These layers will generate contextual embeddings for each token. As the classification head expects a single fixed-size vector, we used mean pooling across the non-padded token outputs which gives one vector that represents the whole sequence. Then this vector is passed to the classification head which will predict the correct answer index(0-3).

During fine-tuning, we experimented training the model for different number of epochs (5, 8, 10, 15) and also tried different learning rates(1e-5, 1e-6, 1e-4) to see which combinations worked best. We trained using the default CrossEntropyLoss[12], and the best-performing model was automatically saved at the end of training based on validation loss. We have also tried freezing some layers of BioGPT to speed up training and reduce overfitting. But, we couldn't freeze the layers because we loaded the model with AutoModel, which doesn't easily expose individual transformer blocks, So we couldn't selectively freeze layers. After running these experiments, we finalized the hyperparameters that consistently gave better results for our classification setup.

Despite experimenting with different maximum sequence lengths, learning rates, and other hyperparameter settings, the overall performance of the model is poor. The classification accuracy was around 25%-27%, which is only slightly better than random guessing. Additionally, we tried using metamap[13] column into the input sequence. But this also did not make any meaningful improvement in the model's performance.

In our experiments, BioBERT achieved a slightly higher accuracy of 28.5%, while BioGPT reached around 26%. This difference is mainly due to the bidirectional architecture of BioBERT, which allows it to better understand the full context of both the question and the answer options. In contrast, BioGPT has a unidirectional architecture, so it struggles to capture the complete context. Despite these slight differences, both models performed poorly overall because our approach of framing the

Table 1. Best Results on BIOBERT for classification task

Class	Precision	Recall	F1
0	0.32	0.28	0.30
1	0.26	0.26	0.26
2	0.30	0.29	0.30
3	0.23	0.27	0.25
Accuracy		0.282	

Total number of correct predictions: 359/1273

Table 2. Best Results on BIOGPT for classification task

Class	Precision	Recall	F1
0	0.18	0.01	0.01
1	0.27	0.35	0.31
2	0.29	0.35	0.31
3	0.20	0.32	0.24
Accuracy		0.25	

Total number of correct predictions: 317/1273

task as a classification problem was incorrect. In a traditional classification problem, the target value remains fixed, but in a multiple-choice format, the target label depends on the position of the correct answer. So, we shifted our approach from classification to a generative modeling strategy.

Phase 2 - Generative Approach

As our approach of framing this task as a classification task is wrong because the target label depends on the position of the correct answer now we changed our approach as Generative approach. And have loaded AutoModelForCausalLM[10] from BIOGPT which has a Language modeling head(Fig: 3) which actually generated the next token after a sequence. We then fine-tuned BioGPT for the generative task, training the model to generate the correct answer token based on the provided question and options.

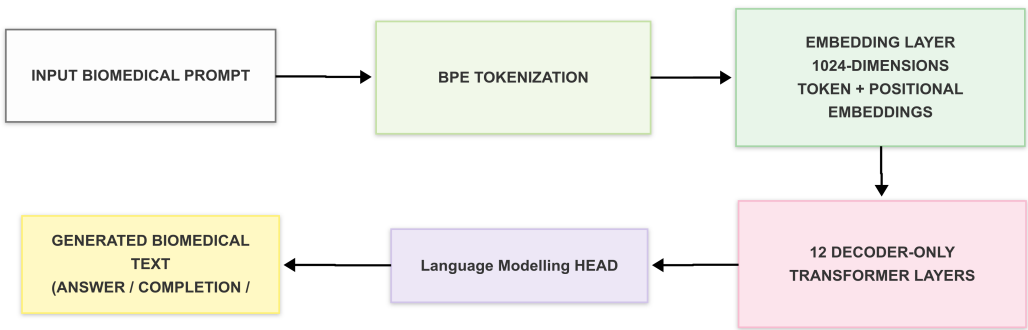

Fig. 3. BIOGPT Architecture for Generative Task

For inputs, we combined the medical question and the four answer choices into a single text sequence as a prompt. We tokenized this prompt using the AutoTokenizer, applying left padding, and experimented with different maximum sequence lengths such as 256, 384, 512, and 600 tokens to find the best one, and finally set it to 512. After tokenization, the input tokens were passed to the model, where each token was first converted into a 1024-dimensional embedding. These embeddings were then processed through the 12 decoder-only transformer layers of BioGPT, which generated contextual embeddings for each token. These contextual embeddings were then used by the language modeling (LM) head to generate the next token. We set the maximum number of generated tokens to 1, so that the model will generate a single output token (the answer). During fine-tuning, we modified the labels and attention masks: we masked all tokens with an ignore index (-100) and set only the label token correctly, allowing the CrossEntropyLoss to compute the loss only based on the generated answer token.

During fine-tuning for the generative setup, we experimented by training the model for different numbers of epochs (3 and 5) and tried different learning rates (2e-5 and 2e-4) to find the best one and set the epochs to 5 and learning rate to 2e-5 and also implemented a learning rate scheduler[12] with a patience of 2 which will reduce the learning rate if the model's validation performance does not improve for 2 consecutive epochs. The AdamW[12] optimizer was used for more efficient weight updates. We trained using the default CrossEntropyLoss while carefully setting the attention masks and labels so that loss was only calculated on the generated answer token. We also attempted partial freezing by freezing all but the top 4 transformer layers of BioGPT to speed up training and improve generalization. However, partial freezing resulted in lower performance (around 26% accuracy), compared to full fine-tuning, which achieved about 32% accuracy.

Table 3. Results of BIOGPT for Generative task

Class	Precision	Recall	F1
0	0.34	0.44	0.38
1	0.30	0.48	0.37
2	0.40	0.17	0.24
3	0.30	0.17	0.22
Overall Accuracy	32.05 %		

Total number of correct predictions: 408/1273

Even though the accuracy and metrics have been improved slightly but the accuracy is still low because the model is generating invalid tokens which is other than 0, 1, 2 and 3. It also might be because the model is not able to understand what to generate. So, we've tried another techniques such as Prompting using Generative models. As BioBERT is not a generative model, we've decided to try Prompting on BioBART(instead of BioBERT) and BioGPT.

Phase 3 - Prompting

After realizing that treating the task as classification was not effective, we shifted towards a generative approach. However, even with generative modeling, we did not achieve good results. So we then moved to prompting techniques, where we would guide the model through careful instructions instead of training it. Since BioBERT is not a generative model and is mainly designed for understanding text rather than generating it, it was not suitable for prompting-based approaches. So, we used BioBART.

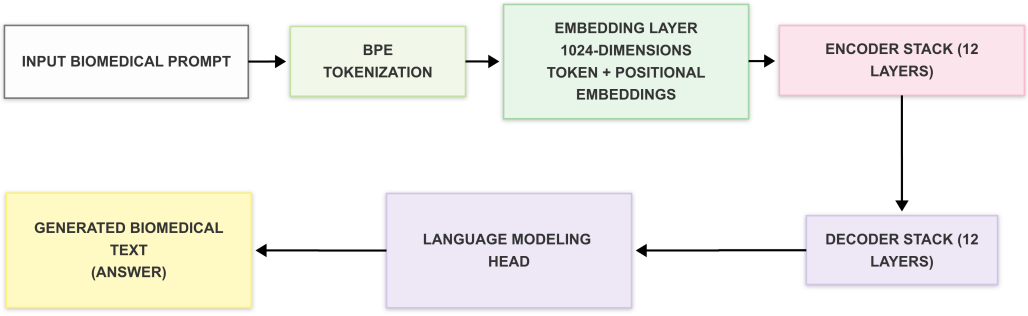

Fig. 4. BioBART Architecture for Prompting

As shown in Figure 4, BioBART uses an encoder-decoder structure where the input text is first processed by the encoder to capture full meaning. The encoder converts the input (question and

options) into a contextual embeddings. These are then passed to the decoder, which generates the output sequence based on the encoded information. This two-step structure helps the model to understand the input and also perform reasoning and text generation. This makes BioBART more suitable for prompting compared to BioBERT.

We first started with zero-shot prompting[5]. In zero-shot prompting, we wrote an instruction that asked the model to read the question and the options, and then pick the correct answer based only on the given information. Below is the format of the prompt we created:

You are a helpful biomedical assistant. Read the following multiple-choice question carefully and select the correct answer based only on the information provided.

Question: [question text]

Options:

A: [option A]

B: [option B]

C: [option C]

D: [option D]

Answer (only generate a single letter from A, B, C, D corresponding to the correct answer):

We used the same prompt for both BioBART and BioGPT. In both cases, we froze the model parameters completely, as there was no training involved. We experimented with different prompts and selected the above prompt because it gave the best results in terms of answer generation and accuracy.

Then, we tried one-shot prompting[5], where we provided the model with one training example along with the test example to help it understand what kind of answer to generate. However, we observed that the model often simply repeated the answer from the training example instead of reasoning based on the test example.

We also tried two-shot prompting[5], but due to the restrictive context window size of 1024 tokens—and because the questions themselves are complex and long—the input was getting truncated. As a result, two-shot and other few-shot prompting methods were not feasible in this case.

Overall, despite all these efforts, it became clear that while prompting was a better strategy than pure classification, the smaller domain-specific models like BioBART and BioGPT still lacked the deep reasoning capacity needed for complex medical multiple-choice questions and also lacked strong instruction-following capabilities. Zero-shot prompting gave relatively the best results among all the methods we tried, but even that could not push the accuracy beyond 30%.

Table 4. Results of Zero-shot Prompting on BioGPT Model

Class	Precision	Recall	F1
0	0.28	0.27	0.27
1	0.28	0.28	0.28
2	0.29	0.27	0.28
3	0.23	0.27	0.24
Accuracy		0.27	

Total number of correct predictions: 344/1272

Table 5. Results of Zero-shot Prompting on BioBART Model

Class	Precision	Recall	F1
0	0.27	0.40	0.33
1	0.27	0.26	0.26
2	0.30	0.18	0.23
3	0.21	0.21	0.21
Accuracy		0.263	

Total number of correct predictions: 335/1272

Additional Techniques:

We also tried several other techniques to improve the model's performance. One approach we tried was log-likelihood scoring. In this method, instead of asking the model to directly generate an answer, we calculated the log-probabilities for each (question, option) pair and selected the option with the highest score. However, even with this scoring-based approach, the accuracy remained low and did not show significant improvement.

We also tried fine-tuning the model first and then applying prompting. Because we thought that additional supervised training on the MedQA dataset could make the models better understand the structure of medical questions before we use prompts to guide them. The models still struggled to perform reasoning and comparing options to the question.

Another idea was to allow the model to generate free-form answers instead of picking from A, B, C, or D directly. After the model generated a text answer freely, we used cosine similarity between the generated text and each of the original options to find the most closely matching option. However, we found that the generated answers were often incomplete, or sometimes closer to wrong options and did not improve the overall performance.

We also experimented with Chain-of-Thought prompting[14], where we asked the model to first do reasoning before giving the final answer but still did not work for this task.

Finally, we tried building an ensemble model by combining the outputs of fine-tuned, prompting, and log-likelihood and generating answer based on majority voting and still did not work for our case.

Despite trying multiple techniques like fine-tuning, prompting, log-likelihood scoring, and even chain-of-thought reasoning, the results remained far below expectations. This is because models like BioBERT, BioGPT, and BioBART, although specialized for biomedical text, had many limitations. Their context windows were relatively small, and they could not properly handle long and complex questions. These models also have relatively few parameters compared to modern large language models (LLMs), limiting their ability to deeply understand complex questions. And they lacked strong instruction-following capabilities and struggled to generate the expected outputs. Moreover, they had difficulty in performing reasoning or comparing the options carefully to the question. On top of these, the questions in the MedQA dataset were extremely complex. Many questions required multi-step reasoning, background medical knowledge, and the ability to compare the options to the questions which are far beyond the capabilities of small domain-specific transformer models.

At this point, we were unsure whether the main problem was with the models capabilities or with the difficulty of the dataset itself. So, we decided to try with using Large Language Models (LLMs). LLMs are trained on massive datasets, have much larger context windows, and are designed to not only understand text but also follow instructions, perform reasoning, and generate thoughtful responses better than small domain specific models.

Phase 4 - Prompting on LLM's

Due to the underperformance of the domain-specific models, we decided to move forward and implement prompting on Large Language Models (LLMs) like GPT-3.5-turbo, GPT-4o-mini, and Gemini 2.0 Flash Lite. Our goal was to check if these LLM's could handle the complexity of the

MedQA task better. We tried only zero-shot prompting because of the limited resources. We have created a very simple and straight forward prompt for the LLMs, where the question and options were given, and the model was asked to respond with only the correct option letter (A, B, C, or D). Below is the prompt format:

"Question is: [question]
Options are:
A. [option A]
B. [option B]
C. [option C]
D. [option D]
Respond with ONLY the correct option letter (A, B, C, or D)."

The same prompt has been used for all the three models. We observed significant improvement in the performance with these models. We got around 59% accuracy for GPT-3.5-turbo, and around 75% accuracy for both GPT-4o-mini, and Gemini 2.0 Flash Lite. This significant improvement is because of the larger context window which helps the model to see the complete question and options without truncation, Pre-trained on much larger corpora which helps in better understanding of text, Stronger reasoning capabilities which helps the model to compare all the options to the question and do better reasoning before generating text and these models also have better instruction following capabilities that helps the model understand what to generate.

Even though LLMs like GPT-3.5-turbo, GPT-4o-mini, and Gemini 2.0 Flash Lite are not specifically trained on biomedical texts, they were still able to perform well. This highlights an important point that while the biomedical background knowledge is helpful, the real challenge in the MedQA dataset lies more in the reasoning complexity of the questions rather than just the domain-specific knowledge. LLMs, with their better training and stronger reasoning skills, were able to handle these complex medical questions better than the smaller biomedical models which we initially used.

Table 6. GPT-3.5-Turbo

Class	Precision	Recall	F1
0	0.58	0.64	0.61
1	0.58	0.63	0.60
2	0.59	0.61	0.60
3	0.63	0.47	0.53
Accuracy		0.59	

Total correct predictions: 708/1200

Table 7. GPT-4o-mini

Class	Precision	Recall	F1
0	0.68	0.82	0.74
1	0.77	0.76	0.76
2	0.76	0.78	0.77
3	0.83	0.61	0.71
Accuracy		0.748	

Total correct predictions: 898/1200

Table 8. Gemini-2.o-flash-lite

Class	Precision	Recall	F1
0	0.75	0.74	0.75
1	0.74	0.74	0.74
2	0.76	0.81	0.78
3	0.76	0.71	0.73
Accuracy		0.754	

Total correct predictions: 899/1200

Discussion on Results

The results clearly show the significant limitations of the domain-specific models we tested, including BioBERT, BioGPT, and BioBART. Across multiple experiments —

Classification on BioBERT and BioGPT (Tables 1 and 2 where both models have low precision and recall which indicates models mostly chose wrong answers and missed the correct ones.

Generative modeling (Table 3) on BioGPT with LM head, where accuracy has slightly improved but precision and recall were still low. The model sometimes guessing correctly.

Zero-shot prompting (Tables 4 and 5) where precision was slightly better for some options, but recall was almost zero for others.

The accuracy on all above experiments is around 25% to 32%. This range is close to random guessing for a four-option multiple-choice task, indicating that these models struggled to perform any meaningful reasoning required by the MedQA dataset. There are many reasons for the underperformance of smaller domain specific models like smaller scale which limits the ability to perform complex reasoning, limited context windows (512 or 1024 tokens) causing truncation of longer questions and options and lacks broader logical reasoning , instruction following capabilities and comparison of options to questions which are needed for answering detailed medical MCQ questions.

In contrast, when we applied simple zero-shot prompting to LLMs like GPT-3.5-Turbo, GPT-4o-mini, and Gemini 2.0 Flash Lite, we observed a major jump in performance (Tables 6, 7, and 8). These models achieved approximately 59% to 75% accuracy without any fine-tuning. Their large scale, larger context windows, and strong instruction-following capabilities helped them to process complex questions fully and reason between options effectively.

Interestingly, the success of LLMs highlights that in the MedQA task, general reasoning ability is more critical than purely domain-specific knowledge. Though LLM's are not pre-trained on biomedical texts, LLMs could outperform smaller specialized models due to their deeper understanding, better reasoning skills, and ability to follow instructions better. Overall, these results suggest that for complex reasoning tasks like medical MCQ questions, using large language models is more effective than smaller, domain-specific transformers.

Conclusion

This project evaluated the performance of smaller domain-specific models like BioBERT, BioGPT, and BioBART and compared them to Large Language Models (LLMs) on the MedQA multiple-choice question-answering task. Initially, we approached the problem as a classification task but later realized that this framing was incorrect, as the target label depends on the position of the correct answer and requires deep reasoning. We also expected that domain-specific models would perform well since they were pre-trained on biomedical text, but they couldn't achieve good performance. We tried several strategies to improve performance, like preprocessing and lemmatization, but, these gave worse results. We also experimented with prompt-based methods such as zero-shot, few-shot, and chain-of-thought prompting. But even these techniques did not help when using the smaller models. This is because the task requires strong reasoning abilities and instruction-following skills to compare the options with the question and generate the correct answer, rather than only domain-specific knowledge. So, we tried zero-prompting on LLMs which performed significantly better because they are capable of understanding complex text, following instructions effectively, and reasoning through multiple steps capabilities that are important for this task.

References

1. Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.
2. Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), September 2022.
3. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
4. Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. BioBART: Pretraining and evaluation of a biomedical generative language model. In *Proceedings of the 21st Workshop on Biomedical Language Processing (BioNLP 2022)*, pages 106–117. Association for Computational Linguistics, 2022. Also available as arXiv:2204.03905 [cs.CL].
5. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020.
6. Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744, 2022.
7. OpenAI. Gpt-4o mini model, 2025. Accessed using OpenAI API.
8. Google DeepMind. Gemini 2.0 flash-lite model. `https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash-lite`, 2025. Accessed via Gemini API on Vertex AI.
9. Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.
10. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020.
11. Ruiqiang Luo, Yankai Lin, Lixin Su, Peng Li, Jinlan Fu, Changjie Fan, Jie Zhou, and Xuanjing Huang. Biogpt: Generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), 2022.

12. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.
13. Alan R. Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.
14. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc V. Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.