A Domain-Specific Approach for Leveraging Language Models in Medical Multiple Choice Question Answering

Shibani Likhite slikhite@ucsd.edu
Savani Suranglikar ssuranglikar@ucsd.edu
Carl Chow cychow@ucsd.edu

1 Introduction

We have witnessed the healthcare field adopt a diverse array of machine learning methods, such as image classification, to aid in patient diagnosis. Additionally, language models have helped doctors and patients to match symptoms with likely underlying causes.

But more importantly, and in contrast to casual consumer chatbots like ChatGPT, the correctness of a model's responses is crucial to the model's performance in the medical field. For instance, we expect an arithmetic error made by ChatGPT to have less negative impact than a medical chatbot suggesting the wrong dosage of medication.

Motivated by the criticality of accurate responses in healthcare, we aim to improve the accuracy of responses returned by language models, specifically for multiple choice question answering in the healthcare domain. This task requires strong reasoning skills and expert domain knowledge.

The medical domain poses unique challenges for QA systems due to its specialized terminology, complex concepts, and vast amount of data, thus making this a non-trivial task. We hypothesize that fine-tuning the model on data of a particular medical subject will enable domain expertise and lead to a better accuracy. In particular, we aim to improve performance on MedMCQA, a multiple choice medical question answering dataset.

By leveraging variant architectures of BERT, we thus aim to tailor the finetuning process to question and answers of a few chosen medical subjects, namely – Anatomy, Psychiatry and Surgery. This approach empowers our models to learn nuanced language patterns specific to each medical subject.

2 What you proposed vs. what you accomplished

In our proposal, we set out five main points to target in our final project. Below we list those points and the progress we made.

1. Acquire and pre-process data ✓

   We were able to load the MedMCQA using Hugging Face's dataset library and preprocessed the entries in a shared Google Colab notebook.

2. Finetune language model on different question categories of the dataset ✓

   We separated the entries of our question dataset into distinct subject categories. Then we finetuned individual models on those questions separately. Our analysis focused on three subjects out of the 21 possible question categories.

   In our project proposal, we mentioned finetuning both transformer-based models and dense passage retrieval models, but given the time constraints, we only experimented with transformer-based architectures such as BERT, BioBERT, and PubMedBert.

3. (Optional) Finetune a question classification model for task ×

   Given the time constraints, we did not train a question classification model, e.g. using BertForSequenceClassification to determine which subject a question belonged to. Rather, we focused on how finetuning a model on individual question categories affects its performance.

4. Analyze the output of the model, do an error analysis ✓

Our group analyzed the incorrect predictions of the model and investigated underlying causes empirically.

5. Work on final reports ✓

3 Related work

In recent years, there has been a considerable effort to construct question answer datasets specifically focused on healthcare. These datasets, including PubMedQA (Jin et al., 2019), CliCR (Šuster and Daelemans, 2018), and COVID-QA (Möller et al., 2020), have gained popularity as standard benchmarks for the field of Medical Question Answering in NLP. However, the healthcare domain encompasses a wide range of intricate medical subjects such as pharmacology, medicine, surgery, etc. There is a scarcity of available datasets when it comes to real-world question answer data categorized based on these complex medical topics. Hence, we have decided to base our experiments on the MedMCQA dataset (Pal et al., 2022) as it is designed to address real-world medical entrance exam questions. The dataset consists of 194k high-quality multiple-choice quetsions MCQs spanning 21 medical subjects in the domain.

Multiple Choice Question Answering (MCQA) tasks in NLP have traditionally been approached as cloze tasks, wherein a portion of the text is removed, and the objective is to predict the missing word or words. However, Robinson et al. (2023) introduces a more natural prompting approach called Multiple Choice Prompting (MCP). Instead of removing text, the question and answer options are presented jointly to the Language Model, which then outputs the symbol (e.g., "A") associated with its chosen answer option. The authors demonstrate that MCP yields state-of-the-art results compared to the conventional Cloze Prompting (CP) method.

The MedMCQA paper (Pal et al., 2022) proposes a similar architecture for multiple choice question answering, employing various versions of BERT (Devlin et al., 2019). The MCQA using BERT involves concatenating the question with each answer option, and the resulting encoding is passed through a feed-forward layer to predict the correct answer option. Among the evaluated variations of BERT, PubMedBERT (Gu et al., 2021) is reported to achieve the highest accuracy, as stated in the provided baseline results.

PubMedBERT is a state-of-the-art, domain-specific pre-trained language model designed for biomedical and healthcare applications. It is trained using BERT, and utilizes a large corpus of biomedical literature, including PubMed abstracts and full texts, for its training data. The paper also presents fine-grained evaluation results per medical subject (eg: Psychiatry, Physiology), which demonstrate better accuracy compared to the results obtained on the entire dataset. These domain specific results have inspired our approach presented in the following section.

4 Your Dataset

We worked with the publicly available MedMCQA dataset (Pal et al., 2022), which contains over 194,000 high-quality AIIMS and NEET PG medical entrance exam multiple choice questions, covering 2.4k healthcare topics and 21 medical subjects are collected with an average token length of 12.77. The train, validation and test split sizes are 182,822, 6,150 and 4,183 respectively.

The dataset emulates the rigor of real word medical exams. We can load the dataset directly from Hugging Face.

Each entry in the dataset comprises eleven columns: the question identifier `id`, the multiple choice question `question`, and four answer options labeled `opa`, `opb`, `opc`, and `opd`. The correct answer is represented as an integer class label `cop`, and the type of multiple choice question `choice_type` as either "single" or "multi". Finally, each entry contains an explanation of the correct answer `exp`, as well as subject and topic names (`subject_name` and `topic_name`) to which the questions are related.

Since MedMCQA was curated with medical question answering in mind, our group did not need to preprocess the dataset to fit our task. In particular, not only is each entry already formatted for a question and answer setting, but we can also use the `subject_name` and `topic_name` fields to train our question classifier. Hence, there is no explicit need for our group to augment MedMCQA for our task.

However, one problem we encountered was that the test set for MedMCQA does not contain the ground truth correct answer responses. In order to test our model performance on the test set, we must submit a `.csv` file of responses to a Google form curated by the original authors of the dataset,

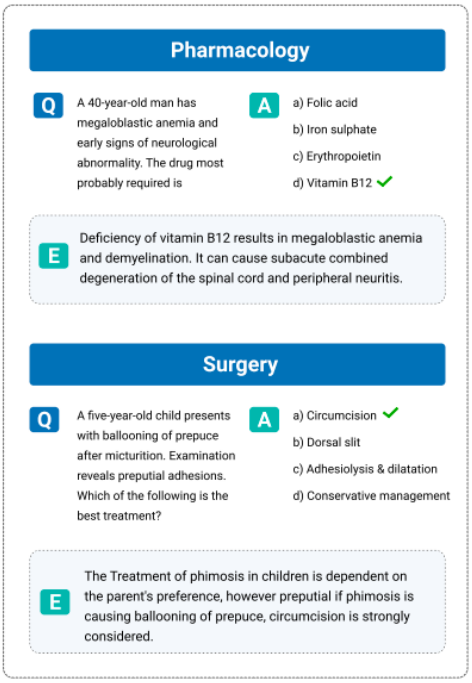

Figure 1: Sample entries from MedMCQA, along with the answer's explanation taken from the paper.

in particular at this GitHub repository. Since our method involves splitting the dataset into subject categories, we instead used the validation set for questions of other categories as our test set. We will explain this approach in more detail in later sections.

4.1 Data Preprocessing

We downloaded the MedMCQA dataset from Hugging Face's dataset library, which was automatically split into train, validation, and test splits. Each entry in the dataset also contains a `subject_name` column, so we further split the train, validation, and test splits into different categories of questions. For instance, we performed most of our analysis on subjects of 'Anatomy', 'Psychiatry', and 'Surgery'.

The original MedMCQA dataset contains separate fields for the questions and four possible responses. As a result, we proprocessed each entry by concatenating the four possible responses to the associated question so that a language model could take the sequences as input.

Furthermore, each question in the training and validation splits of the dataset contains an explanation for the correct answer. As a part of our training step, we occasionally concatenate the explanation of the correct answer to the sequence in addition concatenating the answer options. We will discuss the method which we use to decide when to concatenate explanations in later sections.

—

`CLS` question `SEP` option

`CLS` question `SEP` option `SEP` explanation

—

Figure 2: The two possible input formats into our model. We determine of whether or not to concatenate the explanations of correct answers.

Finally, we tokenized the concatenated responses for each of the subjects that we planned to use for finetuning our model. Hence, our final inputs follow one of the two following formats.

5 Baselines

As our baseline, we calculated the validation accuracy of several transformer-based models from Hugging Face. First, we used an unfinetuned BertForMultipleChoice model configured on 'bert-base-uncased'. Then, we kept the same BertForMultipleChoice model, but configured the weights using BioBERT and PubMedBERT, whose weights we loaded from the Hugging Face library. Since we performed no training on the baseline models, we had no hyperparameters to tune, and simply ran the models on the entire validation set and recorded their accuracies. Again, recall we could not run on the test set because of the lack of correct labels. Also, the Hugging Face loader automatically performs a 80/10/10 split of the data into training, validation, and test sets, so we just kept the default when evaluating the baselines.

For the inference, we pass the concatenated sequence of question and answer option as input to the model, which returns a log-probability of the sequence. Out of the four options, we choose the option that results in the sequence with highest probability as the model's prediction of the correct answer.

Baseline	Accuracy
bert-base-uncased	0.248
BioBERT	0.256
PubMedBERT	0.290

Notice that BERT and BioBERT baselines perform essentially the same as random choice (since we have four answer options). PubMedBERT performed slightly better as a baseline, which can be

attributed to its pretraining on PubMed articles, but still under 30% accuracy.

6 Your approach

6.1 Conceptual Approach

The original MedMCQA paper (Pal et al., 2022) finetuned multiple transformer model architectures on the entire training set and performed inference on the test set.

However, our group wanted to analyze the performance of models finetuned on a subset of questions of the same subject. So after partitioning the dataset into different subjects, we choose which subjects to finetune our models on.

Subject	Train	Validation	Test
Anatomy	14560	234	259
Surgery	16862	369	501
Medicine	17887	295	372

We chose three subjects: Anatomy, Surgery, and Medicine since they were the subjects with the most training data each. We trained our models on the Anatomy section, and tested our implementations on the validation sets of the Surgery and Medicine sections.

We wanted to analyze two main points: whether or not finetuning a model on one subject of questions can perform well on inference on that subject, and whether finetuning a model on one subject of questions still enhances performance on a different subject. Hence, we have a pipeline as follows:

- Choose a question subject from the dataset, e.g. Anatomy.
- Preprocess the data as described in the previous section. In particular, we randomly choose half of the dataset to be tokenized to include the explanation, and the other half of the dataset not to concatenate the explanation. We used this preprocessing method to prevent the model from overfitting to the fact that some explanations start by giving the correct choice.
- Choose a model architecture of BertForMultipleChoice to finetune. Our best experimental accuracies occurred when we used PubMedBert as the basis for our initial weights.
- Finetune the model using the Hugging Face `Trainer` class, on the training set of just questions of our chosen topic, e.g. Anatomy. Note that we do not completely freeze the weights of the BERT layers, so finetuning not only trains the final linear layers, but updates the actual attention layers also.
- Run inference using the finetuned model on the validation set of questions from our chosen subject. We can then update our hyperparameters and rerun the finetuning set for our model.
- Run a performance test of our finetuned model on questions from another subject in our dataset. For example, we finetuned our best performing model on *Anatomy* questions, then updated hyperparameters using the validation set of *Anatomy* questions, and finally ran inference on the validation sets (which we treat as test sets because we did not update hyperparameters based on those subjects) of *Surgery* and *Medicine* related questions.

6.2 Working Implementation

We managed to complete the working implementation to answer multiple-choice questions. We successfully implemented and fine-tuned PubMedBert for the MedicalMCQA task. As proposed earlier, we have fine tuned the model for specific subject-domains. And as hypothesized in the report, we saw an improved accuracy over the baseline by finetuning the model for specific subject domains.

6.3 Other People's Code

To load the dataset from Hugging Face and set up the different splits by subject, we followed the tutorial at Hugging Face Dataset Loading.

To get started with loading the BertForMultipleChoice model, we followed the format of the code on the Hugging Face Multiple Choice page. However, the example Notebook performs inference on the SWAG dataset (Zellers et al., 2018), so we modified many of the data processing functions to work with MedMCQA.

Much of the code in our Notebook was initially modified from the tutorial. Similarly, we repurposed code for inference from CSE 256 Assignment 2.

6.4 Implementation

We have used BertForMultipleChoice, and implemented different pretained models for the task. We experimented around with PubMedBERT, SciBERT and BioBERT. For each of these models, we tried different combinations of hyperparameters and have reported the best accuracy. The complete notebook is submitted using the submission link on Gradescope.

6.5 Compute

We ran the model on a Google Colab Python notebook using a GPU to speed up training, similar to Assignment 2. What was difficult was that the training would stop when the Colab notebook disconnected after a certain period of time. Furthermore, Google would prevent one user from using too much compute time at once. Hence, we saved the model weights for our experiments and would perform inference by loading the save model weights to save time.

6.6 Runtime

Time was an important factor because each model and set of hyperparameters would take between 40 minutes to 90 minutes to train, depending on the number of epochs and batch size. Furthermore, if Colab kicked the notebook off the GPU, the runtime would balloon to ≥ 38 hours. Hence, we trained sparingly and on subsets of the entire dataset.

6.7 Results

We first finetuned a BertForMultipleChoice model using the pretrained weights of PubMedBERT. We used the training set of only *Anatomy* questions, and implemented the training loop using the Hugging Face trainer class. Our best performing model on the *Anatomy* subset was trained with the following hyperparameters:

Hyperparameter	Value
epoch	6
learning rate	$3 \cdot 10^{-5}$
batch size	4
max sequence length	256
weight decay	0.01

Note that max sequence length refers to the maximum number characters that our unprocessed sequence is allowed to have before it is truncated.

Once we have retuned our hyperparameters, we run inference with our model on questions from other subjects. The following are the accuracies for particular subjects:

Subject	Accuracy
Anatomy	0.329
Pathology	0.375
Psychiatry	0.312
Surgery	0.306
Medicine	0.318

In particular, even for subjects that we did not use for finetuning, the model could outperform the baseline and crucially, perform better than random chance. Hence, we observe that the finetuning process also improves the model's overall performance on language tangentially related to the domain on which it was pretrained.

7 Error analysis

Upon analyzing the model's performance, we discovered that it tends to predict answers correctly when the options are shorter in length and easily distinguishable. In the correct example provided, the options are concise and distinct from one another, which likely contributed to the model's accurate prediction. However, the model struggles with samples where the options are longer and share similarities. This is evident in the incorrect example given, where the options are lengthier and resemble each other. Notably, the repetition of the word 'incision' in all options may have confused the model, leading to the incorrect answer.

Correct Prediction Examples:

{

Q: Which of these conditions does not require SABE prophylaxis

a) MR

b) ASD

c) MS

d) CABG

Correct Answer: CABG

Model's prediction : CABG

}

{

Q: In a patient with fresh blow out fracture of the orbit, best immediate management is

a) Wait & watch

b) Antral pack

c) Titanium Mesh

d) Glass bead mesh

Correct Answer: Wait & watch

Model's prediction : Wait & watch

}

Incorrect Prediction Examples:

{

Q:Which one of the following is a muscle splitting incision?

a) Kocher's incision

b) Rutherford-Morrison incision

c) Pfannenstiel incision

d) Lanz incision

Correct Answer: Lanz incision

Model's Prediction: Kocher's incision

}

{

Q:Which one of the following is a muscle splitting incision?

a) Increased ICP

b) Decreased FRC

c) Increased CVP

d) Increased pH

Correct Answer: Increased pH

Model's Prediction: Increased ICP

}

The technique of concatenating the options to the questions and having the model return the log-probabilities of the sequences does perform better than random chance. However, this method still makes the assumption that a factually correct sequence is somehow the more likely to occur in language than a lie or misinformation. Hence, for factually correct multiple choice question answering, a modified method of incorporating more contextual information and outside knowledge may be more suitable than using a language model directly.

8 Contributions of group members

- Carl Chow:

  I worked mostly on setting up the Google Colab notebook, loading the dataset from Hugging Face, and making sure the training and finetuning process for each model worked smoothly. I wrote the code that loaded the pretrained models from Hugging Face, preprocessed/tokenized the data using the methods described previously, and trained the models using the `Trainer` class. I implemented the code for the models to perform inference on the examples, and finally filled out the *Approach*, *Baseline*, and *Data Preprocessing* sections of the report.

- Savani Suranglikar: I researched various datasets related to the task of medical question answering and wrote the related works section. I also worked on the training and finetuning process for the models, particularly finetuning various models for the 'Psychiatry' subject, experimenting with different hyperparameters and running the error analysis and tests. I also wrote multiple sections of our report.

- Shibani Likhite: I worked mostly worked on researching existing implementations on medical Multiple choice question answering. I also trained the model with different hyper parameters and fine tuned it for various subject-domains, experimented around with different models like PubMedBert, SciBert. I wrote the Error Analysis section and other parts of this report along with the Related Works section in the project proposal.

9 Conclusion

Medical question answering has been extensively researched in the field of NLP. Answering entrance-level medical multiple question answers requires domain expertise and sound reasoning. In this project, we have tried to leverage subject-domain expertise and fine-tune the model to improve its performance in a specific subject area. Based on empirical analysis, we have observed that the fine-tuned model shows an improved accuracy when answering questions within that subject. However, it is important to note that multiple-choice question answering remains a relatively new task, and there is still room for improvement in terms of accuracy.

While BertForMultipleChoice is well-suited for predicting the next sentence given a particular sentence, ongoing research aims to develop models specifically tailored for multiple-choice question answering. Moreover, medical question answering presents its own unique challenges due to the intricacies and vast complexity of the domain. One potential avenue for future work involves providing the model with more contextual information, which could enhance its ability to predict answers accurately. For instance, training a separate model on PubMed abstracts and incorporating it to provide additional context may prove beneficial.

In conclusion, our project has demonstrated the potential for subject-specific fine-tuning to im-

prove accuracy in multiple-choice question answering. However, the task as a whole is still evolving, and further improvements are anticipated as researchers continue to explore innovative approaches and incorporate broader contextual understanding into the models.

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W. W., and Lu, X. (2019). Pubmedqa: A dataset for biomedical research question answering. *CoRR*, abs/1909.06146.

Möller, T., Reina, A., Jayakumar, R., and Pietsch, M. (2020). COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Pal, A., Umapathi, L. K., and Sankarasubbu, M. (2022). Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Flores, G., Chen, G. H., Pollard, T., Ho, J. C., and Naumann, T., editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Robinson, J., Rytting, C. M., and Wingate, D. (2023). Leveraging large language models for multiple choice question answering.

Šuster, S. and Daelemans, W. (2018). CliCR: a dataset of clinical case reports for machine reading comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1551–1563, New Orleans, Louisiana. Association for Computational Linguistics.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Conference on Empirical Methods in Natural Language Processing*.