Promptologist : Large Language Model(LLM) for Medical Query Resolution

Aakriti Kinra, Aditi Patil, Aryan Singhal, Lakshay Arora
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{akinra, apatil2, aryans, lakshaya}@andrew.cmu.edu

Abstract

Large Language Models(LLMs) have seen an unprecedented rise in industry and academia, most notably with the advent of ChatGPT. They have found use cases in diverse fields like education, healthcare, software development, design, and sentiment analysis, among others. These models trained on very large data(100s of billions of tokens) and have a very high number of parameters(~ 1.8 trillion for GPT4). Their immense potential comes at the cost of requiring a very large amount of data and compute. Thus their wide development and research has been limited to the biggest tech companies and research labs. Model distillation is a technique that allows us to create smaller, more efficient machine learning models that can achieve high accuracy and performance. Specifically in the field of medicine, we look at MedicalGPT [1] which is a medical QA system. We aim to use the responses generated from MedicalGPT as training data and build a smaller model that can perform just as well. Our smaller LLM, Promptologist is trained on translated Chinese data and English data. The data consists of real-world doctor-patient conversations. We use cross-entropy loss as our loss function and BERT score as our evaluation metric to check the performance of our model.

See Github: Here

1 Introduction

Large Language Models (LLMs) have emerged as a transformative force in artificial intelligence, revolutionizing how machines comprehend and generate human-like text. Deep neural networks power these models, and they show an unparalleled capacity for understanding and generating human language. Some prominent examples include OpenAI's GPT (Generative Pre-trained Transformer) series and BERT (Bidirectional Encoder Representations from Transformers). These models are pre-trained on massive, diverse textual datasets and possess the ability to grasp linguistic nuances, context, and syntactic structures.

LLMs have various applications across industries, showcasing their flexibility and versatility. They are extremely useful in tasks that involve natural language understanding, like sentiment analysis, named entity recognition, and language translation. In the finance sector, large language models are essential for analyzing the sentiment of market news, thus helping investors make informed decisions. They are also used in content creation for generating text that resembles how humans write to cater to diverse needs, from creative writing to content summarization.

In the healthcare sector, large language models have become increasingly important tools, offering significant potential for transformative improvements. LLMs can be utilized for clinical documentation. They can be used for generating a summary of patient records, allowing the doctors and other staff to have a clear and concise narrative of the issue and, hence, speed up the workflow. LLMs also

show promising performance in the field of biomedical research. They can help researchers process and understand large volumes of scientific literature.

Their ability to understand complex medical terminology and extract meaningful information from medical texts makes large language models particularly useful in literature reviews and knowledge synthesis. Since LLMs can analyze a large amount of data, they can be used to examine large amounts of patient records, trial outcomes, and literature. Through this, they can assist in patient care and personalize treatments. They can also identify correlations, predict disease outcomes, and recommend tailored treatment plans. These data-driven insights can revolutionize the accuracy of diagnostics and the efficacy of treatments.

Despite such promising outcomes, the integration of LLMs into the healthcare sector raises some crucial considerations about data privacy, ethical usage, and model interpretability. Ensuring that patient data is handled with the utmost confidentiality and abiding by ethical guidelines is essential for the responsible deployment of LLMs in healthcare.

Now more than ever, there is a growing need to build language models that can resolve patient queries and give accurate and relevant responses. Our project focuses on the development of a specialized Generative Pre-trained Transformer (GPT) tailored exclusively for the medical domain.

Building training inference models is a time-consuming, costly, and resource-intensive task. We want to develop a model that's compact, easy to use, and specifically caters to a particular domain, which in our case is the medical sector. Our model 1 will be designed to answer questions related to medical domains and will be trained on medical dialogue. By addressing these aspects, we create a model that not only fulfills the above requirements but also drives the application of AI in the medical field.

In this report, we examine the prior work in the domain, followed by a discussion of our project. Section 3 explores dataset collection and preprocessing, and Section 4 focuses on the Methodology and Experiments. In the end, we discuss the results, what we conclude from them, and what work we hope to do in the future.

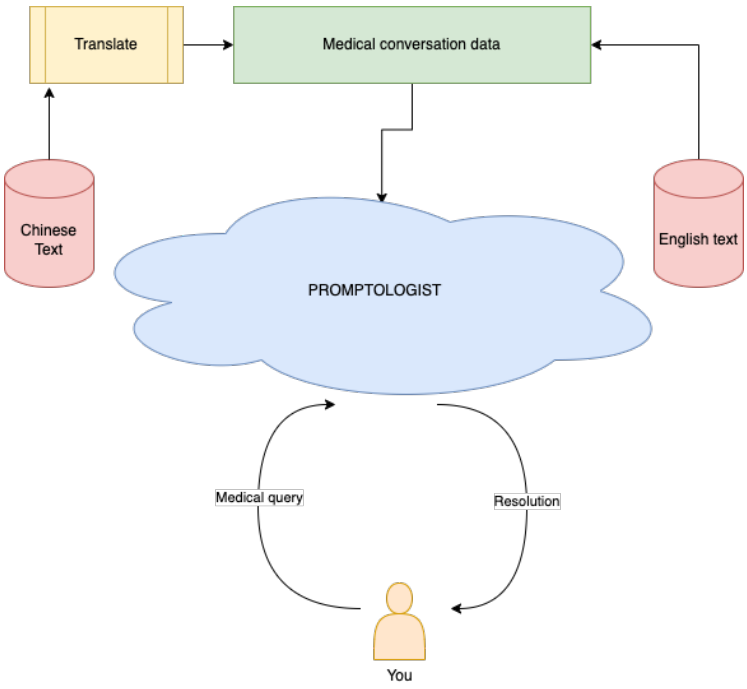

Figure 1: Promptologist Overview

2 Literature Review

The paper [2] provides a comprehensive overview of the use of Large Language Models (LLMs) in the healthcare domain. The authors outline the capabilities of the currently developed LLMs for healthcare and illustrate their development process. The paper provides an overview of the development roadmap from traditional Pretrained Language Models (PLMs) to LLMs. The potential of LLMs to enhance the efficiency and effectiveness of various healthcare applications is explored, highlighting both the strengths and limitations. The authors also conduct a comparison between the previous PLMs and the latest LLMs, as well as comparing various LLMs with each other. The paper investigates the unique concerns associated with deploying LLMs in healthcare settings, particularly regarding fairness, accountability, transparency, and ethics.

The paper [3] presents a comprehensive study on the use of a clinical generative Large Language Model (LLM), GatorTronGPT, in the medical field. The model was trained using a massive dataset of 277 billion words, comprising both clinical and English text. The authors found that GatorTronGPT significantly improved biomedical natural language processing for medical research. Moreover, synthetic NLP models trained using GatorTronGPT outperformed those trained using real-world clinical text. A Turing test conducted by physicians showed no significant difference in linguistic readability and clinical relevance between GatorTronGPT and human-generated text. This paper provides valuable insights into the opportunities and challenges of using LLMs in medical research and healthcare.

MedicalGPT [1] is a new development that brings together natural language processing and healthcare. It involves four key stages: Continued Pretraining (PT), Supervised FineTuning (SFT), Reward Modeling (RM), and Reinforcement Learning (RL). In the Continued Pretraining (PT) stage, the llama-7b model undergoes further pretraining with extensive medical encyclopedia data. This step aims to ingrain domain-specific knowledge into the model, resulting in the creation of the llama-7b-pt model. The next stage, Supervised FineTuning (SFT), involves fine-tuning the pre-trained model using medical question-and-answer data. This process is instrumental in aligning the model with specific medical instructions, culminating in the creation of the llama-7b-sft model. Reward Modeling (RM) introduces a pivotal component by simulating human scoring of text. The project constructs a reward model trained using medical question-and-answer preference data, leading to the development of the llama-7b-reward model. Finally, Reinforcement Learning (RL) leverages the fine-tuned language model (llama-7b-sft) and the reward model (llama-7b-reward) to execute iterative rounds of prompt input, response generation, and policy optimization using Proximal Policy Optimization (PPO). This process results in the creation of the llama-7b-rl model, emphasizing the maximization of output based on the reward model. The project's inclusion of an interactive web interface allows users to seamlessly interact with the model, posing medical queries and receiving informative responses.

The paper Gu et al. (2023)[4] examines a critical aspect of improving the efficiency of complex language models. The researchers discuss an innovative process of knowledge distillation, which involves the transfer of knowledge from a larger model to a smaller model while maintaining or improving the smaller model's performance. They explore the complexities of training the smaller model using both original data and soft labels generated by the larger model. By filtering the essential knowledge from the larger model, the study shows the potential for deploying efficient language models in scenarios where computational resources are limited. This knowledge transfer technique not only enhances computational efficiency but also contributes to the ongoing conversation on resource optimization in the realm of large language models. The paper provides valuable insights for researchers and practitioners looking to balance model performance with computational constraints.

The authors of Gilbert et al. (2023) [5] discuss the potential implications of using artificial intelligence (AI) chatbots as medical devices. They argue that since these chatbots are designed to diagnose or treat medical conditions, they should be subject to the regulations and approval processes required of medical devices. The authors point out that these AI chatbots have the potential to revolutionize healthcare by providing accessible and affordable diagnostic and treatment options. However, they also note that using these chatbots raises significant ethical and safety concerns. For instance, if chatbots are not adequately regulated, they may be used to diagnose or treat conditions incorrectly, leading to serious harm to patients. The authors argue that AI chatbots powered by large language models should be subject to the same rigorous regulatory processes as other medical devices to avoid risks. This would ensure these chatbots are safe and effective before making them available to the

public. The authors also suggest that regulatory agencies should work closely with AI developers to create appropriate regulatory frameworks that ensure the safety and efficacy of these chatbots.

3 Dataset

We will be using the medical dataset used by Medical GPT [1][6][7] that has both Chinese and English data. This dataset was created during the Supervised fine tuning of the MedicalGPT model. It consists of medical queries asked by patients and the response MedicalGPT[1][8] gives to these questions. It has 2.08GB of data with 2.4 million rows of data including the pretrain, finetune and test data.

The MedicalGPT dataset contains real conversations between patients and doctors sourced from different datasets. For the purpose of our baseline, we will only be using the English conversations for training, testing and validation. The data is originally in the form of json objects with input and output being the patient's prompt and doctor's response respectively. We pre-process this data by converting the json into patient-doctor texts.

The task of translating medical conversations from Chinese (Mandarin) to English was a significant challenge due to the specialized nature of the content. Medical terminology is complex and specific, and maintaining the integrity of the information during translation is crucial. Therefore, the selection of the translation tool was a critical decision in our project.

We began by exploring various language translation tools and libraries, focusing on those that were both robust and compatible with our project requirements. Among the libraries we tested were pytranslate[9], translate[10], and googletrans[11]. Each of these tools offered unique features and capabilities, and we conducted extensive testing to assess their performance.

Our evaluation criteria included not only the accuracy of the translation but also the preservation of the medical context. This is particularly important in our project, as the loss of medical context could lead to misinterpretation of the data, potentially impacting the effectiveness of our model.

After rigorous testing and analysis, we decided to use the googletrans library for our data translation. This open-source Python library implements the Google Translate API, which has been proven to be highly effective in maintaining the general meaning in translations. A study conducted by UCLA Medical Center in 2021 found that Google Translate preserved the general meaning in 82.5% of translations, and specifically for Chinese to English translations, the accuracy was 81.7%.[12]

While all the translation tools we tested were able to generate grammatically and semantically correct translations, googletrans stood out for its ability to maintain the medical context of our data. This was a decisive factor in our choice, as preserving the medical context is crucial for the success of our project.

In conclusion, the process of selecting the right translation tool was a meticulous and critical part of our project. The decision to use googletrans was based on its superior performance in our tests, particularly in preserving the medical context, which is vital for our project's success. We believe that this decision will significantly enhance the quality and effectiveness of our model.

4 Model Selection

We conducted experiments using different models: a simple bigram language model, an n-gram model without attention, and an n-gram model with attention. As a baseline, we chose a simple decoder-only n-gram model with attention. Our encoding approach involves character-level tokenization. As part of experimentation we initially attempted word-level tokenization. However, the resulting vocabulary size proved too extensive for our compute. We dropped the proposal of using common NLP preprocessing techniques such as lemmatization, which would result in a reduced dictionary size. However, considering the output requirements of a generative model, need the words in their proper form, and not their lemmatized representation. Yet, upon training for a few epochs, we observed that the generated outputs lacked grammatical correctness in conversational English. Ultimately, character-level tokenization, where each character is encoded with its ASCII value, proved to be the most effective strategy for our scenario. This method, with a smaller vocabulary size, although

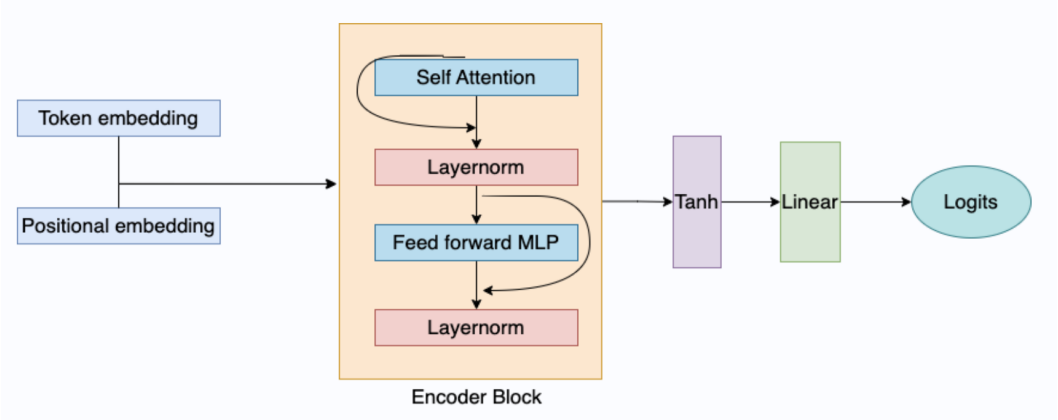

Figure 2: Baseline architecture

a larger timestep dimension, enabled the generation of relevant responses in nearly grammatically correct conversational English.

A schematic of the baseline architecture is given in Fig. 2 showing this. We first transform the prompt with token and positional embeddings. Following this we used transformer encoder blocks as shown in the figure. The transformer block architecture is quite similar to the one used in the original transformer paper [13]. After the input has been processed by transformer encoder block(s), we apply a non-linear activation(Tanh in our case) followed by a linear network to obtain the final logits which are then evaluated by a cross-entropy function.

Further experiments were performed to improve the model and gain more doctor-like responses. CNNs have been used [14] in the past to learn contextually aware feature representations in images and text data as well. GRAM-CNN [15] showed how good results can be obtained by introducing convolution layers for attending to local context in an n-gram embedding for biomedical text. Similar to this work, we explored the idea of adding convolution to strengthen local context during our training process. A similar approach [16] has been shown to learn adequate feature representation for text classification using language modelling. We added the convolution embedding in the transformer encoder block in between the linear layers to enrich the features learned by the model. This is shown in the modified architecture for Promptologist in Fig. 3

Inspired by recent innovations in modifications to the transformer architecture [17, 18], we tried adding LSTMs to the transformer as well. Further ablations included both convolution layers and LSTM layers.

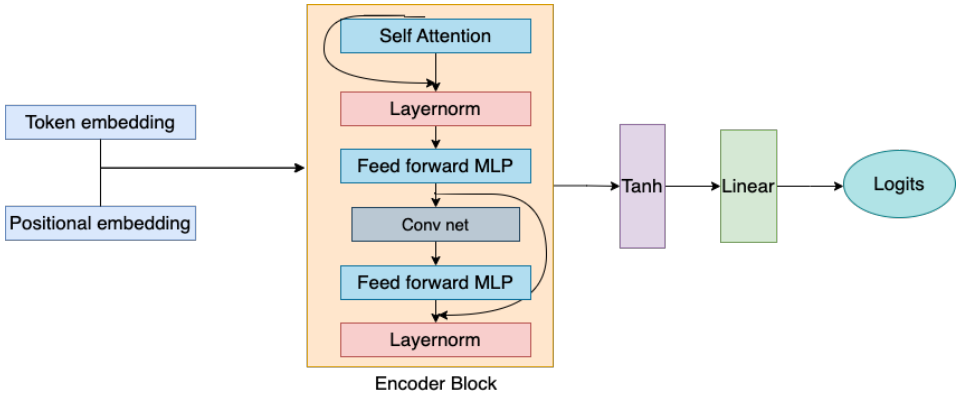

Figure 3: Modified architecture

5 Results

5.1 Loss Function

Cross-entropy loss, or log loss, is a widely used measure in machine learning for evaluating the performance of classification models. It quantifies the difference between predicted probabilities and actual class labels.

The cross-entropy between the true distribution p and the predicted distribution q is given by:

$$H(p, q) = -\sum_{i=1}^{C} p(i) \log(q(i))$$

Where:

- $p(i)$ is the true probability of class i,
- $q(i)$ is the predicted (estimated) probability of class i,
- The sum is taken over all classes C,
- $\log$ is the natural logarithm.

The cross-entropy loss for a single example is given by:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Where:

- y_i is the true label (1 or 0) for class i,
- $\hat{y}_i$ is the predicted probability of class i (output by the model),
- The sum is taken over all classes C.

This loss is typically averaged over all examples in the dataset to get the mean cross-entropy loss. In the binary classification case ($C = 2$), this formula simplifies to the binary cross-entropy loss.

The term "cross-entropy" arises from information theory, measuring the average number of bits needed to encode events from one distribution (true labels) using the optimal code based on another distribution (predicted probabilities). Minimizing cross-entropy equates to improving model predictions, aligning them with the true underlying distribution.

Cross-entropy loss encourages models to assign high probabilities to the correct class, penalizing deviations from the true distribution. It outperforms mean squared error for classification tasks, especially in scenarios where class imbalance is present. Regularization techniques, such as L1 and L2, can be incorporated into the cross-entropy loss to prevent overfitting.

In summary, cross-entropy loss serves as a crucial tool for optimizing classification models, offering a mathematically grounded metric that guides the learning process towards more accurate predictions.

5.2 Evaluation Metric

5.2.1 BERT Score

BERT Score [19] is an advanced metric designed for assessing the quality of generated text in natural language processing tasks. It relies on contextual embeddings from pre-trained models, such as BERT, to intricately evaluate the similarity between reference and candidate sentences, considering both semantic and syntactic aspects.

BERT Score is applicable across various natural language processing domains like machine translation, text summarization, and dialogue generation. It does not solely focus on lexical matching or

n-gram overlap. It understands intricate relationships between words and phrases within the context of the generated text.

The BERTScore computation involves cosine similarity between contextual embeddings generated by BERT for reference ($R = \{r_1, r_2, ..., r_n\}$) and candidate sentences ($C = \{c_1, c_2, ..., c_m\}$). The metric is expressed as:

$$\text{BERTScore}(R, C) = \frac{\sum_{i=1}^{n} \max_{j=1}^{m} \text{BERTSimilarity}(r_i, c_j)}{n}$$

Here, $\text{BERTSimilarity}(r_i, c_j)$ denotes the cosine similarity between contextual embeddings of r_i and c_j obtained from a pre-trained BERT model. The maximum similarity for each reference sentence is selected, and the average is computed over all reference sentences.

The maximum similarity for each reference sentence is selected, and the average is computed over all reference sentences.

BERTScore excels in capturing contextual nuances and semantic similarity, providing a precise evaluation of text quality. Its versatility spans diverse languages and domains without requiring task-specific modifications. Leveraging pre-trained BERT embeddings ensures a thorough understanding of intricate language structures without task-specific training. In conclusion, BERTScore stands out as a technically robust evaluation metric, addressing the limitations of traditional metrics and offering a nuanced perspective on model performance in text generation tasks.

5.2.2 Bleu Score

Within the field of machine translation assessment, BLEU [20] (BiLingual Evaluation Understudy) is a key performance indicator for evaluating machine-translated text. The similarity between a set of excellent reference translations and machine-generated translations is measured by the BLEU score, which runs from zero to one. A score of 1 denotes perfect alignment with the reference translations, indicating high quality, while a score of 0 indicates no overlap between the machine-generated output and the translations, indicating low quality.

The BLEU score is computed using the following formula:

$$BLEU = \text{BP} \times \exp\left(\sum_{n=1}^{N} \frac{1}{N} \log p_n\right)$$

Where:

$$p_n = \frac{\sum_{\text{clipped}} \text{count}_{\text{clipped}}}{\sum_{\text{unclipped}} \text{count}_{\text{unclipped}}}$$

Count-clipped is the count of n-grams in the machine translation that appear in the reference translation, limited to the maximum count in any single reference translation. Count-unclipped is the count of all n-grams in the machine translation.

The Brevity Penalty (BP) is calculated as:

$$\text{BP} = \begin{cases} 1, & \text{len(machine translation)} > \text{len(reference translation)} \\ \exp\left(1 - \frac{\text{length of reference translation}}{\text{length of machine translation}}\right), & \text{otherwise} \end{cases}$$

In practice, BLEU scores are often reported as a percentage (e.g., 0.75 corresponds to 75%). The closer the BLEU score is to 1, the better the machine-generated translation aligns with the reference translations.

On analyzing the results and outputs, we decided to not use BLEU score for our analysis. BLEU score only relies on n-gram matching, which does not capture the overall semantic meaning of a translation. It focuses on local, surface-level similarities and is sensitive to sequence length. It may

not recognize the nuances or fluency of a translation. Thus we use BERT score primarily as a metric of performance.

5.3 Experiments

Table 1: Models Tested

Models	Hidden Layers	Parameters/Hyper-parameters
N-gram with 3layerCNN	3x{Conv1d, ReLU, Dropout}	Number of kernels (64, 128, 256)
N-gram with 3layerCNN	3x{Conv1d, ReLU, Dropout}	Length of kernels (1, 2, 3)
N-gram with 4layerCNN	4x{Conv1d, ReLU, Dropout}	Dropout Values Rate (0.1, 0.2)
N-gram with 1layerLSTM	1 bidirectional LSTM	
N-gram with 2layerLSTM	2 bidirectional LSTM	Length of LSTM (62, 128, 256)
N-gram with 3layerLSTM	3 bidirectional LSTM	
N-gram, 1layerLSTM, 3layerCNN	3x{Conv1d, ReLU, Dropout}, 1 bidirectional LSTM	Number of kernels (64, 128, 256)
N-gram, 2layerLSTM, 4layerCNN	4x{Conv1d, ReLU, Dropout}, 2 bidirectional LSTM	Length of kernels (1, 2, 3) Length of LSTM (62, 128, 256)
N-gram, 3layerLSTM, 3layerCNN	3x{Conv1d, ReLU, Dropout}, 3 bidirectional LSTM	Dropout Values Rate (0.1, 0.2)

In our research, the evaluation of model performance is crucial to understanding the efficacy of our proposed approaches. To this end, we employ the average cross-entropy loss as our primary loss function, providing a measure of the dissimilarity between the predicted and actual outputs. Additionally, we utilize the BERT score as our chosen evaluation metric, as it offers a more nuanced assessment by considering contextual representations and employing cosine similarity, aligning with our objective of generating outputs that capture the essence and relevance of the patient's prompt.

Initially, we experimented with the BLEU score as an evaluation metric. However, we quickly recognized its limitations in our context, as it emphasizes exact matches between predictions and outputs. Given our goal of producing outputs that are similar in essence and contextually relevant, we opted for the BERT score, which better aligns with the intricacies of our task.

Our baseline model, employing an n-gram approach with multi-head attention, yields promising results, achieving a precision of 0.7781, a recall of 0.8055, and an F-1 score of 0.7916 (as detailed in the table below). Subsequent explorations involved various architectural modifications, including the incorporation of convolutional neural networks (CNNs) and long short-term memory networks (LSTMs).

Through extensive experimentation, we identified that the n-gram model with multi-head attention, complemented by a 3-layer CNN, exhibited superior performance. This configuration yielded a BERT Precision score of 0.7924, a BERT Recall score of 0.8286, and a BERT F1 score of 0.8101. These results showcase the effectiveness of the selected architecture in capturing contextual nuances and generating outputs that align closely with our evaluation criteria.

Interestingly, while alternative architectures, such as the n-gram model with multi-head attention and a 3-layer LSTM, demonstrated lower average cross-entropy values, they did not fare as well on our BERT evaluation metric. This underscores the importance of aligning the evaluation metric with the specific goals of the task, as not all models that excel in one metric necessarily perform optimally in others.

Overall, our findings highlight the significance of leveraging the BERT score as a comprehensive evaluation metric and the success of the n-gram model with multi-head attention and a 3-layer CNN in achieving our desired performance outcomes. The detailed results, including precision, recall, and F1 scores, are presented in the accompanying table for reference and further analysis.

Table 2: Average Cross Entropy Loss and BERT score of Testing Set

Models	Avg. Cross Entropy Loss	BERT Precision	BERT Recall	BERT F-1
bi-gram	2.6472	0.4282	0.4567	0.4420
n-gram w/o attention	1.8298	0.6622	0.6582	0.6602
n-gram with attention	1.3273	0.7781	0.8055	0.7916
n-gram with attention & LSTM	0.0878	0.7249	0.7872	0.7547
n-gram with attention & CNN	1.1765	**0.7924**	**0.8286**	**0.8101**
n-gram with attention & CNN & LSTM	**0.0664**	0.7539	0.7971	0.7749

6 Conclusion

Our research has been dedicated to addressing the pivotal challenge of size reduction in Language Model (LLM) architectures, with a dual focus on maintaining performance excellence while significantly mitigating computational demands. The strategic choice of a Transformer encoder as our foundational architecture, coupled with inventive modifications, has been instrumental in our pursuit. Notably, our experimentation unveiled that the integration of multi-head attention with convolution layers as an embedding mechanism stands out as a promising solution, demonstrating superior results in terms of efficiency and effectiveness.

Moreover, our findings underscore a fundamental characteristic of language models—the inherent capacity to leverage task-domain knowledge. This intrinsic ability empowers these models to generate outputs that closely resemble human-like responses, thereby enhancing their practical utility across diverse applications.

In a broader context, our work extends beyond the realms of academic research and contributes to the real-world application of natural language processing. The impact of our endeavors is particularly noteworthy in the domain of medical consultations and extends to various other fields. By successfully reducing the computational burden without compromising performance, our approach holds the potential to revolutionize how language models are employed in resource-intensive tasks.

7 Future Work

Looking ahead, our vision involves further refining our model's performance through an expansion of training data in terms of quantity and diversity. Augmented training strategies will be implemented, and we will continue to explore optimal n-gram sizes, seeking a balance between precision and recall for our n-gram model. Addressing challenges posed by unseen test data will involve the exploration of diverse smoothing techniques.

Our future research trajectory encompasses the fine-tuning of existing models to assess their efficacy in capturing medical context within conversational scenarios. Additionally, we plan to leverage medical databases and literature to enhance our model with medical knowledge, enabling it to provide more adept responses to a variety of prompts beyond its initial training data.

To adapt our model to user-specific needs, a feedback system will be implemented, collecting user input to refine and tailor the generated responses over time. This adaptive approach is crucial for aligning the model's outputs with user expectations and evolving requirements. Ultimately, we anticipate the potential for these models to serve as valuable tools for on-the-spot consultations, offering insightful information before engaging with a professional medical expert.

8 Division of Work

Aakriti Kinra

- Data collection, pre-processing and translation
- Performed experiments on the model and ran ablations
- Documentation including report and presentation

Aditi Patil

- Data collection, data pre-processing, and data translation
- Performed experiments on the model and ran ablations
- Documentation of report, presentation, and Github repository

Aryan Singhal

- Creation of baseline and ablations with CNNs
- Data cleaning, pre-processing, and translation
- Documentation, report writing, presentation and team management

Lakshay Arora

- Creation of baseline and new model architectures using CNNs and LSTMs
- Performed experiments on the model and ran ablations
- Data pre-processing and translation
- Documentation including report and presentation

References

[1] Ming Xu. Medicalgpt: Training medical gpt model. `https://github.com/shibing624/MedicalGPT`, 2023.

[2] Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv preprint arXiv:2310.05694*, 2023.

[3] Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. A study of generative large language model for medical research and healthcare. *arXiv preprint arXiv:2305.13523*, 2023.

[4] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

[5] Stephen Gilbert, Hugh Harvey, Tom Melvin, Erik Vollebregt, and Paul Wicks. Large language model ai chatbots require approval as medical devices. *Nature Medicine*, pages 1–3, 2023.

[6] Chinese medical dialogue data. `https://github.com/Toyhom/Chinese-medical-dialogue-data`.

[7] Huatuo-26m. `https://github.com/FreedomIntelligence/Huatuo-26M/blob/main/README_zh-CN.md`.

[8] Jianquan Li, Xidong Wang, Xiangbo Wu, Zhiyi Zhang, Xiaolong Xu, Jie Fu, Prayag Tiwari, Xiang Wan, and Benyou Wang. Huatuo-26m, a large-scale chinese medical qa dataset, 2023.

[9] Jjangsangy. py-translate, 2015. Software available from `https://pypi.org/project/py-translate/`.

[10] Unknown. translate, 2021. Software available from `https://pypi.org/project/translate/`.

[11] Suhun Han. googletrans, 2020. Software available from `https://pypi.org/project/googletrans/`.

[12] Phrase. How accurate is google translate? 2023 performance report, 2023. Available from: `https://phrase.com/blog/posts/is-google-translate-accurate/`.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] Arik Poznanski and Lior Wolf. Cnn-n-gram for handwritingword recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2305–2314, 2016.

[15] Qile Zhu, Xiaolin Li, Ana Conesa, and Cécile Pereira. GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, 34(9):1547–1554, 12 2017.

[16] H. Wang, J. He, X. Zhang, and S. Liu. A short text classification method based on n -gram and cnn. *Chinese Journal of Electronics*, 29:248–254, 2020.

[17] DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. Block-recurrent transformers, 2022.

[18] Ciprian Chelba, Mohammad Norouzi, and Samy Bengio. N-gram language modeling using recurrent neural network estimation, 2017.

[19] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[20] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.