

Creating a medical chatbot that answers questions accurately using an RAG technology.

Report by Vikas Vyas

Introduction

Large language models (LLMs) have become a global household staple in the last few years. Most popular one being ChatGPT. LLMs are advanced Natural Language Processing (NLP) models that are trained on billions of tokenised texts using a deep learning (DL) architecture 1. LLMs like Chat-GPT generate text when given a specific prompt. These models are capable of handling tasks like summarisation, translation, conversation, image generation and interpretation, problem-solving, solving math questions, and more 2.

Question-answering, especially in biomedical space, has been traditionally managed by Information Retrieval (IR) systems like PubMed. However, IR is just presenting relevant documents for further reading. With the advancement of NLP and DL models like Bidirectional Encoder Representations from Transformers (BERT), Embeddings from Language Models (ELMo), and now surpassed BiDAF (Bi-Directional Attention Flow), Question-answering has been revolutionised 3.

One such use-case is in healthcare. In the UK, half of the adults find it difficult to access healthcare information they can trust and 1 in 10 adults have become prey to misinformation 3. To tackle these challenges, a trustworthy LLM that can answer medical questions accurately will provide a perfect solution. However, literature search suggests that most medical Question-Answering systems are designed towards healthcare professionals and not patients 4,5.

This report aims to analyse the performance of four LLMs, Meta Llama 3, Google Gemma, Google T5, and Microsoft Phi-4-mini, on answering medical questions presented by patients. The answers from LLMs will be analysed against reference answers scraped off the National Health Service (NHS) UK website. Three evaluation metrics will be used, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores, BERT scores, and human evaluation for robustness.

The dataset is synthetically generated using ChatGPT coming under the OpenGPT model developed by a team of researchers from University College London and King's College London. It is also human validated by an expert in clinical field.

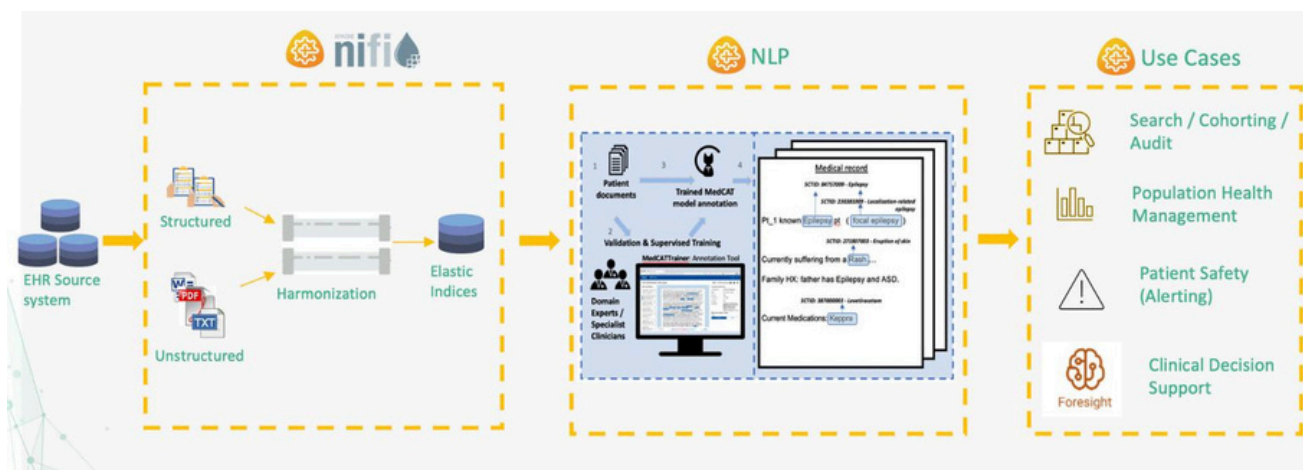


Figure 1: The architecture of OpenGPT. 6

Dataset has two components: training file and test file. The training file has 24,005 question-answer pairs for 2,392 diseases. The test file has 211 question-answer pairs.

index	question	answer	disease	reference_url
0	What is bronchiolitis?	Bronchiolitis is a common chest infection that...	Bronchiolitis	https://www.nhs.uk/conditions/Bronchiolitis/
1	What are the early symptoms of bronchiolitis?	The early symptoms of bronchiolitis are simila...	Bronchiolitis	https://www.nhs.uk/conditions/Bronchiolitis/
2	What should I do if my child is having trouble...	If your child is having difficulty breathing, ...	Bronchiolitis	https://www.nhs.uk/conditions/Bronchiolitis/

Table1: training datatop3 rows. The test data has the same columns. The question column contains a patient question or query regarding a disease. Answer contains the answer scraped from the NHS website. Disease has the assigned disease the question is regarding.

Reference_url is the NHS online source the information is taken from .

A folder named `search_data` contains the reference information about the 2,392 diseases in separate .txt files.

The test file will be utilised for model performance checks. The test file has an average word length of 7 for questions and 42 for answers, while the maximum question is 18 words long and answer is 110 words long.

Methodology

Preprocessing

Preprocessing is done to decrease the size of the dataset, select only relevant features, remove duplicates, and handle or impute any missing values. It also involves tokenising text, word embedding, lemmatisation, and stemming.

The dataset required very little preprocessing in the means of cleaning, and imputation as it had no missing values or differently formatted values. The *answer* and *question* word length were within expected range.

The text doesn't need to undergo lemmatisation and stemming since these LLMs, Phi-4, Gemma-3, and Llama-3, are pretrained on unprocessed natural language and would lose contextual embedding when lemmatized or stemmed.

Tokenization is the act of turning text into its most frequent consecutive pairs of words or characters, called symbols ⁷. Essentially, tokenization is the act of text compression using an algorithm. The four LLMs have their respective tokenizers like Byte Pair Encoding (BPE) for Phi-4 or SentencePiece for Gemma, T5 and Llama, and don't require additional tokenisation on the text itself. BPE replaces the original text with symbols hence reducing the text size.

$$nndnnndn \rightarrow YdYndn \rightarrow Zndn$$

$$nndnnndn \rightarrow nYnnYn \rightarrow ZnZn \rightarrow XX$$

Equation 1: BPE demonstration using a random sequence of characters. The first equation replaces nn as Y symbol and then Yd and Z. In second equation, which is more efficient, nd is replaced by Y, nY as Z, and Zn as X.⁷

SentencePiece tokenizer builds upon the BPE and uses a probabilistic way of tokenizing on raw sentences with whitespaces and Unicode characters. SentencePiece also adds multilanguage support and lossless compression⁸.

The disease text files in the *search_data* folder were named using the following code:

```
def safe_filename(name):
    return re.sub(r'[^w\-\_\. ]', '_', name).strip().replace(" ", "_") + ".txt"
```

The above code allows the files to be named conventionally and replace non-alphanumeric characters with an underscore and have a .txt in the end as an extension. Lastly, only in T5 LLM, the training dataset was used to train the model. The training file has limited usability for the other LLM models in this report as they are decoder-only models using only prompts and hyperparameter tuning.

Algorithm Design

The four LLM models work on different architectures and designs. However, Llama, Gemma, and Phi4 work on same decoder-only transformer model. Whereas T5 works on an encoder-decoder model.

Models →	MetaLlama3-8B	Gemma 3-7B	Phi-4-mini-instruct	T5- Sm all
Parameters↓				
Parameters (B = Billion, M = Million Tokens)	8B	7B	3.8B	220M
Training data Size (T = Trillion tokens)	15T SentencePiece	6T	7T	220M
Tokenizer	Pre-trained decoder-only Transformer	SentencePiece	BPES	SentencePiece
LLM Architecture	Pre-trained decoder-only Transformer	Pre-trained decoder-only Transformer	Pre-trained decoder-only Transformer	Pre-trained Encoder-Decoder Transformer

Table 2: Comparison of the 4 LLM models on various parameters

The four models here were used in a pipeline to generate answers for the test file and then compare it with reference answers already present in the test file. Three approaches were used for Llama 3, Gemma 3, and Phi-4: (1) Comparing generated responses from LLM answers with reference answers as a baseline. (2) Retrieval Augmented Generation (RAG) with Term Frequency–Inverse Document Frequency (TF-IDF), and (3) RAG TF-IDF with GloVe (Global Vectors for Word Representation) Embeddings.

For T5, only the first approach was used due to compatibility issues with GloVe embeddings while testing. The BERTScore and ROUGE metrics were drastically lower than Approach 1 without GloVe. Hence the metrics were dropped due to possible technical errors. RAG was not used since the objective is to test the T5's performance on test set after training it on a training dataset.

RAG is an IR algorithm that retrieves the reference information from external sources for better accuracy of the generated text. The TF-IDF is a method of text vectorisation that assesses the frequency and importance of words in a contextual manner 9.

The retrieval is based on Facebook AI Similarity Search (FAISS) tool which vectorises documents for faster search. FAISS uses K-nearest neighbours and based on cosine similarity, returns the $k = n$ nearest vectors or documents 10.

Lastly, GloVe is used for word embeddings in a multi-dimensional vector space to retain contextual meaning and map similarity between words. E.g., ice and water are placed closer in the vector space 11.

Metrics and Hyperparameters

Three metrics, Rouge, BERTScore, and Human evaluation are utilised to assess the performance of the four LLM models.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a simple evaluation strategy which compares the n-gram, or word units generated artificially that match exactly with the human generated text. Here, ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-L sum were calculated. ROUGE-1 and ROUGE-2 are unigram and bigram calculations respectively. ROUGE-L is the longest subsequence match score, and ROUGE-L sum is like ROUGE-L but works on a sentence level instead of text at whole. ROUGE is a recall-based metric 12.

BERTScore relies on contextual embedding vectors from BERT LLM model to determine cosine similarity between generated and reference tokens. Precision, recall, and F1 scores are shown for BERTScore 13.

Last metric was Human Evaluation. Although, not as scientific, as the other two, human evaluation relies on checking the quality of generated text compared to reference texts. Out of 100 a score is provided for each model. Deductions are made on non-standard text formatting, grammatical mistakes, incorrect disease categorisation, irrelevance and incoherence of generated answers, and lastly unempathetic messaging. Since, the question-answer system or chatbot is designed for patients, it needs to be empathetic in its messaging. This was done for 25 randomly selected generated and reference answers.

The output of Llama 3, Gemma 3, and Phi-4 was entirely determined by the prompt which was consistent for all three models. For T5, the output was determined by the training.

For hyperparameter tuning, only focus was paid to temperature. Temperature allows the model to adopt a deterministic or probabilistic next-word generation approach. Deterministic outputs have a temperature of 0.0 or <3 and predict the highest probability word as the next word. Probabilistic output has a higher temperature and results in a more varied response that lacks context compared to deterministic approach. For this report, temperatures of 0.7 and 0.1 were used. Model performance at 0.7 temperatures were comparable to 0.1 (± 0.1 unit change in metrics) and hence, the 0.7 temperature results were excluded from the report for redundancy reasons. For all three models using RAG TF-IDF pipeline, the k was kept constant ($k = 5$) which considers the number of nearest or most similar documents.

Replicate and Huggingface API keys were used to access Llama 3, and Gemma 3 models respectively.

Results

The four models were compared across three metrics, ROUGE, BERTScore, and human evaluation. The results are presented in three tables. Results in Table 3 are based on Approach 1 where only LLM generated responses are compared with the reference answers. Results in Table 4 are based on Approach 2, where LLM generated responses with RAG and TF-IDF pipeline are compared with reference answers. Approach 3 results, represented in Table 5, had the additional GloVe embeddings on top of Approach 2.

F1 scores of 0.7 or above and ROUGE-L score of 0.5 or above are considered good. These will be thresholds for this report's assessment.

Establishing a baseline is tricky for this report since T5 model has not been applied to Approaches 2 and 3. And Gemma-3 model has performed poorly on all Approaches.

Model	ROUGE				BERTScore			Human Evaluation
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L sum	Precision	Recall	F1-Score	
T5	0.27	0.13	0.23	0.23	0.68	0.48	0.53	60
Gemma-3	0.09	0.33	0.01	0.13	0.07	0.25	0.13	30
Phi-4	0.32	0.12	0.23	0.25	0.65	0.42	0.53	65
Llama-3				0.24	0.58		0.62	

Table 3: Approach 1, without RAG. Straightforward comparison of generated responses from the LLM with the reference answers from the test file.

For Approach 1, the results indicate lower overall scores across all models. However, Phi-4 has the best performance with F1-score of 0.65 and ROUGE-L score of 0.25. Under human evaluation, the Phi-4 scores better than all models due to its accurate answers. However, Phi-4 does not give compassionate answers in this approach. Llama-3 provides longer, more contextual answers. But the answers themselves are not significantly better than Phi-4's. However, Llama-3 displays better comprehension. Gemma-3 performs poorly compared to similar models Phi-4 and Llama-3 and even worse than T5. The responses by Gemma-3 have non-standard formatting, use markdown language (despite prompting it not to!) and provides inaccurate information for most diseases and short descriptions instead of elaborating in some cases.

Model	ROUGE				BERTScore			Human Evaluation
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L sum	Precision	Recall	F1-Score	
T5	X	X	X	X	X	0.48	X	X
Gemma-3	0.09 0.47	0.01 0.30	0.07 0.41	0.07 0.41	0.42 0.76	0.68	0.46 0.71	35 80
Phi-4	0.56	0.41	0.49	0.50	0.76	0.76	0.76	90
Llama-3								

Table 4: Approach 2, with RAG. Comparison of generated responses from the LLM with reference answers from the test file. RAG searches the search_data directory with disease text files with extra information.

Under Approach 2, the scores are expected to be better than Approach 1, due to the robustness of recall from source files provided by RAGs. In Table 5, except Gemma-3 model, Phi-4 and Llama-3 have improved their scores significantly. Llama-3 is the best performing model across all metrics. Noticeably, the ROUGE scores for Llama-3 are almost 10 decimal points better than Phi-4.

Model	ROUGE				BERTScore			Human Evaluation
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L sum	Precision	Recall	F1-Score	
T5	X	X	X	X	X	0.42	X	X
Gemma-3	0.07 0.46	0.08 0.30	0.05 0.40	0.06 0.40	0.39 0.75	0.68	0.40 0.71	30 75
Phi-4	0.55	0.40	0.48	0.48	0.76	0.76	0.75	87
Llama-3								

Table 5: Approach 3, RAG with GloVe embedding. Approach 2 with additional GloVe embeddings.

Lastly, with GloVe embeddings, the scores are expected to improve even further with additional contextual and semantic clarity provided by GloVe. However, for Phi-4 and

Llama-3 models, the ROUGE scores went 1-2 decimal point lower than Approach 2. BERTScores have remained almost the same between Approach 2 and 3 for Phi-4 and Llama-3. For Gemma-3, the scores have gone down by few decimal points across all metrics except Rouge-2 between Approaches 2 and 3.

For Human evaluation across Approaches 2 and 3, the results were almost identical for Phi-4 and Llama-3. However, integration with RAG resulted in lower coherence and lack of empathetic language in some responses. Gemma-3 featured the most empathetic language out of all the models.

DiscussionandConclusion

Answering patient queries is a challenging task, especially when the generated responses need to be based on accurate information from verified sources. An attempt to create a patient question-answering tool was undertaken in this report comparing different LLM models.

The best performing model was the Meta Llama-3 followed by Microsoft Phi-4-mini across most metrics and across all Approaches. Third would be the T5 model and worst performing model was Google Gemma-3. Gemma-3 performs inadequately across all metrics, and the quality of generated responses is poor. This is most likely due to poor compatibility with the code used for Phi-4 and Llama-3. Use of Huggingface pipeline for Gemma-3 instead of Replicate is also noteworthy.

In recent literature, ChatGPT-4 has the best performance answering patient questions and is considered the gold-standard or the benchmark to beat. Llama models also have consistently high performance scores for patient question-answering for multiple studies 14,15. T5 model has consistent performance (comparable ROUGE scores to this study) across one paper found for mental health patient question-answering 16. However, for the Phi-4 and Gemma models, relevant literature was not found for patient question-answering tasks.

Overall, Llama-3 is an excellent model to use for patient question-answering tasks and breaches the thresholds for good performance metrics (ROUGE > 0.5 and BERTScore > 0.7). The same is confirmed via multiple studies using the Llama-3 model, however, the studies usually use the bigger token version of the models. Addition of RAG and GloVe embeddings also significantly boosted model performance making it more adherent to medical guidelines.

With better tweaking of some hyperparameters, better performance can be achieved.

Limitations

Due to resource and computing restraints, the smaller models have been used which do not perform as well as the bigger token size versions of the same models. Also, a cross-validation pipeline for the same would have made the results more robust.

Lack of proper code integration for the Gemma model is acknowledged in this report. Efforts to redo the models with consistent code will be of high priority moving forward.

References

1. Blank, I. A. What are large language models supposed to model? Trends in Cognitive Sciences 27, 987–989 doi:10.1016/j.tics.2023.08.006 (2023).
2. Naveed, H. et al. A Comprehensive Overview of Large Language Models. Preprint at <https://doi.org/10.48550/arXiv.2307.06435> (2024).
3. Farrington, S., Randall, S., Campbell, S., West, R. & Exton-Smith, H. PIF and Ipsos publish Knowledge is Power. Patient Information Forum <https://pifonline.org.uk/resources/knowledge-is-power/> (2024).
4. G, K. et al. Question answering systems for health professionals at the point of care-a systematic review. Journal of the American Medical Informatics Association: JAMIA 31, doi:10.1093/jamia/ocae015 (2024).
5. Singhal, K. et al. Toward expert-level medical question answering with large language models. Nat Med 31, 943–950 doi: 10.1038/s41591-024-03423-7 (2025).
6. CogStack. CogStack Product Documentation - CogStack Documentation - Confluence. <https://cogstack.atlassian.net/wiki/spaces/COGDOC/pages/2406875137/CogStack+Product+Documentation>.
7. Kozma, L. & Voderholzer, J. Theoretical Analysis of Byte-Pair Encoding. Preprint at <https://doi.org/10.48550/arXiv.2411.08671> (2024).
8. Kudo, T. & Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Preprint at <https://doi.org/10.48550/arXiv.1808.06226> (2018).
9. Lewis, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Preprint at <https://doi.org/10.48550/arXiv.2005.11401> (2021).

10. Douze, M. et al. The Faiss library. Preprint at <https://doi.org/10.48550/arXiv.2401.08281> (2025).
11. Pennington, J., Socher, R. & Manning, C. Glove: Global Vectors for Word Representation. in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Association for Computational Linguistics, Doha, Qatar, 2014). doi:10.3115/v1/d14-1162.
12. Lin, C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. in Text Summarization Branches Out 74–81 (Association for Computational Linguistics, Barcelona, Spain, 2004) <https://aclanthology.org/W04-1013/> .
13. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q. & Artzi, Y. BERTScore: Evaluating Text Generation with BERT. Preprint at <https://doi.org/10.48550/arXiv.1904.09675> (2020).
14. Chen, H., Fang, Z., Singla, Y. & Dredze, M. Benchmarking Large Language Models on Answering and Explaining Challenging Medical Questions. arXiv <https://arxiv.org/html/2402.18060v1> (2024).
15. Forkum, J. D. et al. Comparative evaluation and performance of large language models on expert level critical care questions: a benchmark study. Critical Care 29, 72 doi: 10.1186/s13054-025-05302-0 (2025).
16. Maurya, R., Rajput, N., Diviit, M. G., Mahapatra, S. & Ojha, M. K. Exploring the potential of lightweight large language models for AI-based mental health counselling task: a novel comparative study. Sci Rep 15, 22463 doi: 10.1038/s41598-025-05012-1 (2025).