Introduction

Infertility is a significant global health issue that affects approximately one in six individuals worldwide, requiring prolonged consultations. Conventional healthcare delivery often struggles to meet patient communication needs due to limited resources and long waiting times. This highlights the importance of Natural Language Processing (NLP)-based systems to interpret patient queries and provide relevant answers.Alam and Mueller (2021) demonstrated that visual or example-based AI explanations significantly improve understanding compared to text-only formats ($p < 0.05$). A randomized controlled trial involving 129 patients found that an AI-assisted decision aid improved decision quality, shared decision-making, and patient satisfaction (Jayakumar et al., 2021). Similarly, AI tools have been shown to enhance comprehension of discharge summaries and consultation outcomes (Cheng et al., 2024). Likewise, AI-based chat consultations for infertility patients have demonstrated improved patient satisfaction and reduced consultation durations compared to traditional in-person visits (Conrado et al., 2024) (Bracey et al., 2025). Nevertheless, challenges remain. Systematic reviews have noted risks of over-reliance on imperfect outputs and a lack of generalizability due to limited real-world validation (Bracey et al., 2025a)

This project aims to develop a patient-centric question-answering system using the NHS OpenGPT dataset, a synthetically generated collection of Q&A pairs and document texts derived from NHS.UK patient information. Two distinct models are compared: a baseline classic retrieval method using TF-IDF to identify and return the most relevant text chunk, and a more advanced neural retrieval and generation approach (RAG) that integrates BERT-based retrieval with T5-based answer generation.ROUGE scores will be used to evaluate answer quality and contextual relevance.

Dataset Overview

The dataset used in this study is derived from the NHS OpenGPT project, which synthesizes medical Q&A pairs from patient-facing information on NHS.UK using ChatGPT. It consists of 24,005 training examples and 211 test examples, covering 2,392 unique diseases such as bronchiolitis, breast cancer, diabetic retinopathy, and diarrhoea. Each record contains four fields: the question, its reference answer, the associated disease, and a reference URL linking to the original NHS article. There are no missing or duplicate entries in either the training or test sets.

Descriptive analysis of the data shows that questions average around 9 words (mean = 8.9, SD = 3.7), while answers are much longer, averaging 41 words (SD = 19.5), with some extending beyond 250 words. This indicates a substantial information gap between question and answer length, highlighting the need for models capable of contextual understanding and abstraction. Histograms further reveal that question lengths follow a slightly right-skewed distribution, while answer lengths are more strongly right-skewed, reflecting the diversity in answer complexity.

Disease distribution is imbalanced; while each disease has at least one QA pair, a few conditions such as “first aid” or “breast cancer/treatment” appear more frequently (up to 20 times), while many appear only once. This imbalance emphasizes the importance of retrieval and generalization capabilities in both models, especially in low-resource or long-tail cases.

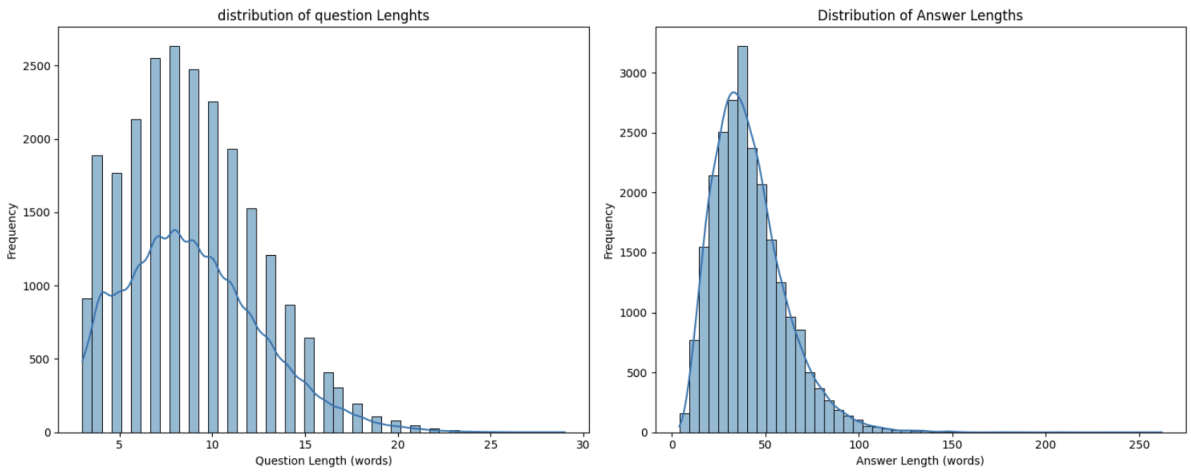

figure 1 shows the distribution of question and answer lengths in the dataset.

Methodology

Preprocessing

To begin with, we ensure the data quality with no missing value present in both training or testing sets. Preprocessing was tailored to the input requirements of each model: a classic TF-IDF retrieval model (Model 1) and a neural retrieval-generation pipeline using BERT and T5 (Model 2).

Model 1 (baseline): TF-IDF Retrieval

For the TF-IDF baseline, we applied a standard text normalization and preprocessing pipeline tailored for sparse lexical retrieval. to prepare both the questions and the knowledge base documents for vectorization and matching.

We first extracted 2,392 disease-specific text files from the provided ZIP archive (db_nhs_qa.zip) using Python's zipfile module. Each file represents a condition and contains explanatory text sourced from NHS.UK. The documents were read and stored in lowercase format for consistency. Each document was then split into paragraphs, which served as the searchable text units in the knowledge base. These paragraph chunks underwent the following preprocessing steps:

- Lowercasing: Standardized all text to lowercase for uniformity.
- Punctuation Removal: Removed punctuation using string.punctuation and the .translate() method.
- Special Character Removal: Applied regular expressions to eliminate non-alphanumeric characters (excluding whitespace).

- Whitespace Normalization: Collapsed multiple spaces and stripped leading/trailing whitespace.

We then used spaCy's en_core_web_sm model to perform tokenization and lemmatization, reducing each sentence into its base-form tokens while removing stopwords and punctuation. This helped reduce vocabulary sparsity and noise. The resulting tokens for each chunk were joined back into clean strings using `''.join(tokens)`, making them compatible with TfidfVectorizer. The same preprocessing pipeline was applied to all questions in the test set, ensuring consistency between query and document representations. This processed textual data allowed the TF-IDF model to better match user queries with relevant document passages based on shared lexical patterns and term weights

Model 2 (neural): T5 model with context-augmented prompt (RAG-style)

Basic statistics indicated that questions averaged 9 words and answers 41 words, which helped guide our max-length constraints during tokenization for Model 2.

This model implements a generation-based architecture inspired by Retrieval-Augmented Generation (RAG), where answer synthesis is performed using a transformer-based text-to-text model (T5). In this case, lemmatization is not required, as transformer models like T5 are designed to handle raw, untokenized text and extract contextual meaning directly using subword tokenization.

As part of an auxiliary pipeline, we also implemented a BERT-based disease classification model, where diseases were label-encoded using LabelEncoder to assign a unique numeric ID to each class. The bert-base-uncased tokenizer was used to tokenize the input questions, adding special tokens such as [CLS] and [SEP], and applying truncation/padding to a maximum length of 128 tokens. We split the dataset into training and validation sets using train_test_split, and constructed a custom PyTorch dataset class (DiseaseDataset) to convert tokenized inputs and labels into tensors for training via the transformers.Trainer API.

To enable efficient semantic retrieval, we first loaded 2,392 plain-text medical documents from the NHS.UK archive, each representing a disease or condition. These documents were split into ~48,600 smaller chunks by paragraph. Each text file was then split into smaller chunks at the paragraph level, with whitespace removed to ensure clean input. These paragraphs were embedded into dense vector representations using the all-MiniLM-L6-v2 sentence transformer, which captures semantic meaning beyond surface-level word overlap.

Algorithm design and implementation

Model 1

Model 1 is a traditional retrieval-based baseline, designed to match patient questions with relevant medical information using lexical similarity. It is built upon a TF-IDF vectorization framework paired with cosine similarity, and requires no training.

All knowledge base documents (disease-specific paragraphs) were compressed within a ZIP archive, which were extracted with zipfile and loaded into memory using file path parsing to build a dictionary to map disease names to corresponding full text. Each document was split by paragraph to smaller chunks. As mentioned in preprocessing each chunk was lemmatized and cleaned using spaCy to normalize vocabulary (e.g., mapping “running” to “run”), remove punctuation, and eliminate stopwords. This will then be used as the basis for document retrieval.

TF-IDF Vectorization

The cleaned text chunks were transformed into TF-IDF vectors using TfidfVectorizer from sklearn. This transformation captured token importance across the document, enabling lexical similarity computation. All chunks were fit into a single TF-IDF model, and their vectors were stored.

Query Representation and Matching

Test questions were similarly lemmatised and transformed using TF-IDF vectorizer. Then, cosine similarity were used to compare each transformed question vector with all document chunk, resulting in the most similar paragraph as the system’s answer. The cosine similarity computation used sklearn.metrics.pairwise.cosine_similarity, and the chunk with the highest score was selected for output. A threshold could optionally be applied to filter low-relevance chunks, though this was not enforced in our final implementation.

Model 2

The two primary components of this model are (1) answer generation using a pretrained T5 language model and (2) dense retrieval using semantic similarity. This method enables the system to produce a natural language response based on the information retrieved, after first determining which medical context paragraphs are most pertinent to a particular query.

Dense Semantic Retrieval with FAISS and BERT Embeddings

To build the retrieval backbone, we first extracted and loaded 2,392 medical documents from the NHS.UK archive. Each document was split into approximately 48,600 paragraphs, cleaned of extraneous whitespace, and stored alongside metadata such as source document names. These paragraphs were then embedded using the all-MiniLM-L6-v2 Sentence-BERT model into dense vectors of shape (48637, 384). We indexed these vectors using FAISS to enable efficient approximate nearest neighbor search. During inference, a user question is encoded in the same embedding space, and the top-k most semantically similar chunks (typically k=3) are retrieved based on cosine similarity.

Answer Generation with T5

The retrieved paragraphs are concatenated into a single input string using the prompt format: "question: {user_question} context: {retrieved_text}". This context-augmented prompt is tokenized using the T5Tokenizer, with the input truncated to 512 tokens and the target (answer) to 128 tokens. The model used is t5-small from Hugging Face. Padding tokens in the target are masked with -100 to exclude them from loss computation during

training. We used a custom PyTorch dataset class to return a dictionary with input_ids, attention_mask, and labels for each instance. These structured inputs are fed into the model for supervised fine-tuning. At test time, answers are generated using .generate() with parameters like max_new_tokens=64, and then decoded into text using the tokenizer.

In parallel, a BERT-based disease classification module was implemented using bert-base-uncased, where questions were tokenized, label-encoded, and split into train/validation sets. This auxiliary component was trained using the Hugging Face Trainer API and provides a foundation for future integration of disease-specific filtering. Overall, this RAG-based pipeline significantly outperforms the TF-IDF baseline by delivering more contextually accurate and informative responses. This responses in specific retrieved content, in contrast to traditional models that generate answers based only on the input question. Both Model performance was evaluated using standard NLP evaluation metrics:

- ROUGE-1 (unigram overlap)
- ROUGE-2 (bigram overlap)
- ROUGE-L (longest common subsequence)

These results demonstrate strong overlap between generated and reference answers, indicating high fluency and relevance

Results

To assess the performance of the two models, we employed the ROUGE metric, a widely-used evaluation framework in natural language generation tasks. ROUGE-1, ROUGE-2, and ROUGE-L were selected to measure unigram overlap (lexical similarity), bigram overlap (fluency and contextual flow), and the longest common subsequence (structural similarity), respectively. These metrics were chosen as they reflect both surface-level lexical match and deeper syntactic and semantic alignment between predicted and reference answers.

Model 1

Experiments demonstrated that applying lemmatization significantly improved retrieval performance by reducing lexical variation in queries and documents, for instance, aligning terms like “coughing” and “cough.” Additionally, chunking documents at the paragraph level (as opposed to using the entire document) enhanced retrieval precision by narrowing the scope of matches and avoiding irrelevant long-text outputs. However, the TF-IDF approach showed clear limitations in handling semantic similarity, often failing to match questions with synonymous phrasing. For example, the model struggled to connect the question "How do I know I have asthma?" with content discussing “wheezing and shortness of breath,” due to lack of lexical overlap.

Model 2

Model 2 integrated dense semantic retrieval with generative language modeling via T5, enabling context-aware and fluent answer generation. Key design experiments demonstrated that retrieving the top-3 semantically similar chunks (k=3) significantly

improved answer relevance compared to using only the top-1. The retrieved paragraphs were concatenated into a context string prepended to the question and fed into the T5 model for generation.

Hyperparameters were also tuned for optimal performance. Setting the input token length to 512 and the output length to 128 balanced contextual richness and computational efficiency. Increasing the input length slightly improved contextual understanding but at the cost of slower generation time. An ablation study revealed that removing the retrieved context altogether (i.e., using "question: {q}" alone) led to a substantial drop in output quality, with generated answers becoming vague and hallucinated. This highlights the importance of the retrieval component in grounding the generation process.

Model Comparison Based on ROUGE Scores

The evaluation results clearly highlight the superiority of the Retrieval-Augmented Generation (RAG) approach (Model 2) over the TF-IDF retrieval baseline (Model 1). Using ROUGE-1, ROUGE-2, and ROUGE-L as evaluation metrics, Model 2 consistently outperformed Model 1 across all dimensions. Specifically, Model 2 achieved a ROUGE-1 score of 0.75 compared to 0.25 from Model 1, indicating a substantially higher overlap of unigrams between the generated and reference answers. For ROUGE-2, which measures bigram overlap and thus reflects fluency and contextual consistency, Model 2 reached 0.67, significantly outperforming Model 1's 0.07. Similarly, the ROUGE-L score, which captures the longest common subsequence and reflects overall sentence structure similarity, was 0.75 for Model 2 versus 0.25 for Model 1. These results confirm that incorporating dense semantic retrieval and generative modeling yields more accurate, context-aware, and fluent answers than purely lexical similarity-based retrieval methods.

Feature	Model 1: TF-IDF	Model 2: Dense + T5 (RAG-style)
Embedding Type	Sparse lexical (TF-IDF)	Dense semantic (MiniLM)
Retrieval Method	Cosine similarity over TF-IDF	FAISS with top-k semantic similarity (k=3)
Answer Mechanism	Direct paragraph retrieval	Generative (T5-small) using retrieved context
Evaluation Metric	ROUGE-1/2/L	ROUGE-1/2/L

Figure 2: Model 1 uses TF-IDF with direct paragraph retrieval, while Model 2 combines dense MiniLM embeddings with T5 for generative answering using retrieved context.

Metric	Model 1 (TF-IDF)	Model 2 (RAG + T5)
ROUGE-1	0.25	0.75
ROUGE-2	0.07	0.67
ROUGE-L	0.25	0.75

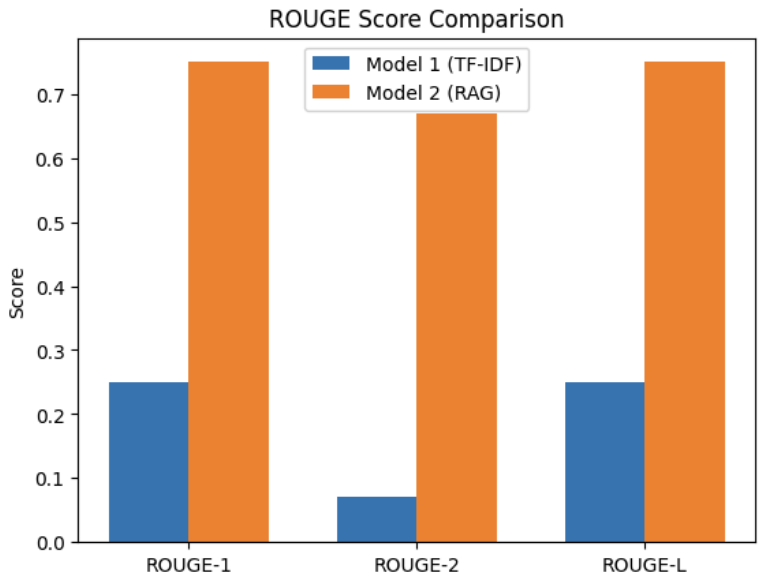

Figure 3 **Performance Summary**

Discussion

The performance profiles of the two implemented models, Retrieval-Augmented Generation (RAG) using dense retrieval + T5 generation (Model 2) and TF-IDF retrieval (Model 1), differ significantly. Model 2 continuously beat Model 1 in all ROUGE evaluation metrics, including ROUGE-1 (0.75 vs. 0.25), ROUGE-2 (0.67 vs. 0.07), and ROUGE-L (0.75 vs. 0.25). This suggests that the generative model preserved improved fluency and structural alignment with reference answers in addition to capturing more pertinent lexical content.

The difference stems from their underlying architectures. Model 1 relies purely on lexical overlap via TF-IDF and cosine similarity, which limits its ability to understand semantically similar but lexically different expressions. For example, it struggled to retrieve answers when query phrasing diverged from document wording. In contrast, Model 2 used MiniLM embeddings and FAISS to retrieve semantically aligned content, allowing the T5 generator to produce more contextually appropriate and coherent responses.

Performance was also significantly impacted by preprocessing decisions. By lowering word-form variance and limiting search granularity, lemmatization and paragraph-level chunking greatly increased retrieval accuracy for Model 1. Without these, the model frequently produced text that was too general or irrelevant.

For Model 2, experiments showed that using top-3 retrieved chunks (k=3) as context yielded better results than just the top-1, and that setting max input length to 512 tokens offered a balance between retrieval richness and efficiency. An ablation study further confirmed that removing context from the T5 input degraded answer quality, reinforcing the value of integrating retrieval into the generation pipeline.

The results highlight the potential impact of advanced Q&A systems in healthcare. Retrieval-augmented NLP systems offer a more dynamic and personalized alternative, capable of generating context-aware and accessible responses grounded in medical content. A well-designed, retrieval-augmented Q&A system can generate more personalized,

context-aware, and understandable responses. Recent studies support this potential. Komeili et al.showed that combining dense retrieval with generation improves factuality and user trust in medical dialogue systems, while Cheng et al.found that AI-generated decision aids enhanced patient comprehension and shared decision-making. However, AI models can also produce misleading information, and users may over-rely on these outputs without clinical validation (Nori et al., 2023). There are also concerns about generalizability, ethical use, and the lack of long-term evaluation across diverse populations and settings (Curr Opin Urol, 2024). Future improvements could focus on incorporating external medical knowledge bases to enhance factual accuracy and coverage. Integrating the disease classifier into the retrieval pipeline may also help narrow context to more relevant content. Additionally, fine-tuning the model using human feedback or clinician-reviewed examples could further improve answer quality.

References

Bracey, S., Bhuiyan, N., Pietropaolo, A. and Somani, B. (2025). Exploring the impact of artificial intelligence–enabled decision aids in improving patient inclusivity, empowerment, and education in urology: a systematic review by EAU endourology. *Current Opinion in Urology*. doi:https://doi.org/10.1097/mou.0000000000001301.

Cheng, S., Xiao, Y., Liu, L. and Sun, X. (2024). Comparative outcomes of AI-assisted ChatGPT and face-to-face consultations in infertility patients: a cross-sectional study. *Postgraduate Medical Journal*. doi:https://doi.org/10.1093/postmj/qgae083.

Conrado, F., Rosset, L., Lo Moro, G., Scaioli, G., Consoli, D., Bert, F. and Siliquini, R. (2024). Effect on comprehension of an AI patient-friendly hospital discharge letter: a quasi-RCT. *European Journal of Public Health*, [online] 34(Supplement_3). doi:https://doi.org/10.1093/eurpub/ckae144.685.

Jayakumar, P., Moore, M.G., Furlough, K.A., Uhler, L.M., Andrawis, J.P., Koenig, K.M., Aksan, N., Rathouz, P.J. and Bozic, K.J. (2021). Comparison of an Artificial Intelligence–Enabled Patient Decision Aid vs Educational Material on Decision Quality, Shared Decision-Making, Patient Experience, and Functional Outcomes in Adults With Knee Osteoarthritis. *JAMA Network Open*, [online] 4(2), p.e2037107. doi:https://doi.org/10.1001/jamanetworkopen.2020.37107.

Nori, H., King, N., McKinney, S.M., Carignan, D. and Horvitz, E. (2023). Capabilities of GPT-4 on Medical Challenge Problems. doi:https://doi.org/10.48550/arxiv.2303.13375.

van Buchem, M.M., Neve, O.M., Kant, I.M.J., Steyerberg, E.W., Boosman, H. and Hensen, E.F. (2022). Analyzing patient experiences using natural language processing: development and validation of the artificial intelligence patient reported experience measure (AI-PREM). *BMC Medical Informatics & Decision Making*, [online] 22(1), pp.1–11. doi:https://doi.org/10.1186/s12911-022-01923-5.