Group 9- Project Report: Healthcare Answer Summarization

Riya Gupta (2022410) || Swara Parekh (2022524) || Vimansh Mahajan (2022572)

Abstract

Healthcare-related answers often encompass detailed explanations, making it difficult for users to identify key information quickly. This project aims to develop a perspective-aware summarization system that uses natural language processing (NLP) techniques to generate concise and contextually relevant summaries of medical answers. The system is designed to preserve the original intent and perspective. The generated summaries are trained and evaluated using BLEU and BERTScore as the primary evaluation metrics.

1 Introduction

In the digital age, individuals increasingly turn to online platforms for healthcare information. However, the answers they receive—whether from medical professionals or community forums—are often lengthy, complex, and filled with diverse perspectives. This makes it difficult for users to quickly grasp the essential insights they need.

This project aims to develop a Natural Language Processing (NLP) system that can automatically generate concise, informative, and perspective-aware summaries of healthcare-related answers. By identifying and preserving key perspectives such as *information*, *suggestions*, *experiences*, *causes*, and *questions*, the system will provide clear and structured summaries that retain the original intent of the response.

Such a system has the potential to support both healthcare professionals and the general public in efficiently understanding medical discussions, improving accessibility and clarity in health communication.

The organization of the paper is as follows: Section 2 discusses *Related Works*; Section 3 highlights the *Methodology*, including the pipeline and the main models' implementation. The model experiments are outlined in Section 4. The results obtained and their analysis are presented and discussed in Section 5. In Section 6, the *Conclusion and Future Work* are highlighted.

2 Related Work

2.1 Perspective-aware Healthcare Answer Summarization

Link to Paper

The paper introduced **PUMA**, a novel dataset for perspective-specific summarization of healthcare-related community question-answering (CQA) threads. The dataset comprises 3,167 medical questions and ∼10,000 answers, annotated with five distinct perspectives: *Cause*, *Suggestion*, *Experience*, *Information*, and *Question*. The authors proposed **PLASMA**, a summarization model built on the Flan-T5 backbone with prefix-tuning. PLASMA incorporates a novel energy-controlled loss function that combines perspective-specific, tone-specific, and anchor-specific energy scores.

PLASMA outperformed five strong baselines (including Flan-T5, BART, GPT-2, and PEGASUS) across multiple metrics. It achieved ROUGE-L F1 of 21.38, BERTScore of 0.869, and BLEU of 0.0405.

2.2 Comparative Analyses of Transformer Models for Text-Based Emotion Recognition

Link to Paper

The paper focuses on comparing four transformer-based models—**BERT**, **RoBERTa**, **DistilBERT**, and **XLNet**—for emotion recognition using the International Survey on Emotion Antecedents and Reactions (ISEAR) dataset. The models classify text into seven emotion categories: *Anger*, *Disgust*, *Sadness*, *Shame*, *Fear*, *Joy*, and *Guilt*. All models demonstrated effectiveness in emotion detection, with **RoBERTa** achieving the

highest accuracy among them.

3 Methodology

3.1 Dual-Head Classifier for Perspective Detection and Span Tagging

To jointly detect perspectives and identify relevant spans in answers, we design a dual-head classifier architecture.

The input to the model is a concatenation of the question and answer. A shared encoder (BioBERT) processes the input to produce contextualized token embeddings.

- **Perspective Classification Head:** A multi-label classification head is applied to the [CLS] token to predict which of the five perspectives (INFORMATION, SUGGESTION, EXPERIENCE, QUESTION, CAUSE) are present. It uses a sigmoid activation for independent binary decisions over each class.
- **Span Tagging Head:** A token-level classification head performs BIO-style tagging for each token in the answer span, with 11 possible labels: O, B-*perspective*, and I-*perspective* for each perspective type. This head predicts which parts of the answer correspond to each perspective.

The model is trained using a combination of binary cross-entropy loss (for the classification head) and token-level cross-entropy loss (for the tagging head). This setup allows the classifier to both detect the presence of each perspective and highlight the corresponding text spans.

The output of this classifier is used in the generator stage to guide perspective-controlled summarization.

3.2 Fine-Tuning a BART Generator with LoRA

To generate summaries aligned with specific perspectives, we fine-tune a BART-based sequence-to-sequence model using Low-Rank Adaptation (LoRA), which allows efficient fine-tuning with a reduced number of trainable parameters.

Input Construction: Each input to the model includes the context, question, and extracted answer spans from the original QA pair. The text is structured as follows:

```
Context: <context>
Question: <question>
Answers: <answer span>
```

The output consists of one summary for each of the five perspectives (INFORMATION, SUGGESTION, EXPERIENCE, QUESTION, CAUSE), formatted as:

```
<PERSPECTIVE> SUMMARY: <summary>
```

Model Architecture: We use `facebook/bart-large-cnn` as the base model, augmented with LoRA adapters injected into attention layers. These adapters are trained while keeping the original BART weights frozen, enabling efficient adaptation with minimal computational overhead.

Training Setup:

- The model is trained using a cross-entropy loss over all five perspective summaries concatenated into a single output.
- LoRA parameters are configured with rank $r = 8$, $\alpha = 32$, and a dropout rate of 0.1.
- We use a batch size of 4, gradient accumulation of 2 steps, and train for 10 epochs.

Evaluation: After training, the model generates summaries for each individual perspective by prepending a directive such as:

```
Generate a <PERSPECTIVE> summary:
<input>
```

The outputs are evaluated per perspective using both BLEU and BERTScore metrics. Reference summaries are parsed from the target text and compared with the generated outputs.

Advantages:

- LoRA makes fine-tuning efficient even on large models like BART.
- Perspective-specific prompting enables controllable and targeted summarization.
- Evaluation per perspective ensures fine-grained quality assessment.

3.3 Fine-Tuning BART with LoRA on Hard Examples

To further enhance the performance of the model, we fine-tune the previously trained BART-based generator with LoRA adapters on a dataset consisting of hard examples. These examples are those for which BERT score < 0.84, to challenge the model, helping it generalize better by learning from more complex input-output pairs.

Input Construction: The input format consists of the context, question, and extracted answer spans from the original QA pair. The input text is structured as:

```
Context: <context>
Question: <question>
Answers: <answer span>
```

The output consists of one summary for each of the five perspectives (INFORMATION, SUGGESTION, EXPERIENCE, QUESTION, CAUSE), formatted as:

```
<PERSPECTIVE> SUMMARY: <summary>
```

Model Architecture: We use the `facebook/bart-large-cnn` model as the base, augmented with LoRA adapters injected into the attention layers. The LoRA configuration is set with rank $r = 8$, $\alpha = 32$, and a dropout rate of 0.1. Only the LoRA adapters are trained, while the base BART model weights are frozen, allowing for efficient fine-tuning with minimal computational overhead.

Training Setup: The model is fine-tuned on a dataset of hard examples, which are preprocessed and formatted into consistent input-output pairs. The training parameters include:

- A batch size of 4, gradient accumulation over 2 steps, and training for 3 epochs.
- The model is trained using cross-entropy loss over the concatenated five perspective summaries.
- The model's LoRA weights are loaded from pre-saved `adapter_model.safetensors` and incorporated into the model.

Evaluation: After fine-tuning, the model generates summaries for each perspective, which are evaluated using BLEU and BERTScore metrics. The reference summaries are parsed from the target text, and the generated outputs are compared for quality.

3.4 Pipeline Code for Final Evaluation

The final evaluation pipeline involves integrating both the classifier and generator models to process test examples and generate perspective-based summaries. The pipeline is outlined below.

Constants and Mappings: We begin by defining essential constants for perspectives (INFORMATION, SUGGESTION, EXPERIENCE, QUESTION, CAUSE), as well as BIO tag mappings used for span-based tagging. Mappings for perspectives to IDs and vice versa, and BIO tags to IDs, are also initialized.

Perspective	Precision	Recall	F1-score	Support
INFORMATION	0.875	0.947	0.910	735
SUGGESTION	0.894	0.934	0.914	595
EXPERIENCE	0.850	0.899	0.874	316
QUESTION	0.855	0.461	0.599	102
CAUSE	0.650	0.561	0.602	139

Table 1: Results of the Classifier Model

Utility Functions: Utility functions are defined for loading JSON data, joining multiple answers into a single string, and extracting reference summaries from the test data.

Classifier Model: A dual-head classifier model is used, which incorporates a pre-trained RoBERTa model. The classifier has two heads: one for predicting the presence of perspectives and another for span tagging. The model is initialized, loaded with pre-trained weights, and set to evaluation mode.

Generator Function: The generator function utilizes a pre-trained BART model to generate summaries for each of the five perspectives. A prompt is constructed dynamically based on the input text and the target perspective. The model is used to generate and decode the summary for each perspective.

Evaluation Function: The evaluation function computes BLEU and BERTScore metrics for the generated summaries by comparing them against reference summaries. These metrics are computed for each perspective individually.

Main Pipeline: The pipeline first loads both the classifier and generator models. It then iterates through the test data, applying the classifier to predict which perspectives are present and subsequently using the generator to create summaries for those perspectives. If no perspectives exceed a threshold, summaries for all perspectives are generated.

Steps of the Pipeline:

1. Load the classifier and generator models.
2. Process each test example:
    - Classifier predicts the relevant perspectives for each example.
    - Generator produces summaries for the predicted perspectives.
3. Store the results, including predicted perspectives, generated summaries, and reference summaries.

4. Evaluate the generated summaries using BLEU and BERTScore metrics.

3.5 Genereated Summaries File

To facilitate training and evaluation, each data sample is represented as a JSON object containing the following fields: the input `question`, the corresponding `answer`, the predicted `perspectives` (a list of one or more of the five categories: Information, Suggestion, Experience, Question, Cause), and the generated summaries (`generated_summaries`) for each perspective. Additionally, reference summaries (`reference_summaries`) curated by human annotators are included for evaluation.

An example format is shown below (truncated for readability):

```
{
  "question": "I am extremely tired and
      slowly gaining weight...",

  "answer": "it sounds a lot like
      hypothyroidism...",

  "predicted_perspectives": ["
      INFORMATION", "SUGGESTION", "CAUSE
      "],

  "generated_summaries": {
    "INFORMATION": "You don't need to
        diet to lose weight...",
    "SUGGESTION": "All that needs to be
        done is balancing the input...",
    "CAUSE": "The average woman needs
        100 grams of lean protein..."
  },

  "reference_summaries": {
    "INFORMATION": "Hypothyroidism,
        characterized by insufficient
        thyroid...",
    "SUGGESTION": "While being active is
         beneficial, it's advised to
        consult...",
    "CAUSE": "Low energy and weight gain
         may be due to hypothyroidism
        ..."
  }
}
```

This structure enables the model to perform multi-perspective summarization and allows for fine-grained evaluation using metrics such as ROUGE, BLEU, and BERTScore across individual perspectives.

4 Experimental Setup

4.1 Dataset

We use a modified version of the PUMA (Perspective sUMmarization dAtaset) dataset (naik2024perspective), originally developed for perspective-aware summarization in the healthcare domain. The original dataset comprises 3,167 medical question-answer threads sourced from the healthcare category of Yahoo! Answers, with over 9,000 community-generated responses. Each thread is annotated at two levels: (1) answer spans labeled with one or more semantic perspectives, and (2) corresponding abstractive summaries per perspective.

In our work, we modify the original dataset to better suit our pipeline architecture. Specifically:

- We **merge the *Clarification* perspective into *Information***, following the original authors' own annotation refinement process, due to significant semantic overlap.
- We **remove the *Treatment* perspective** entirely, to avoid potential ethical and medical liability issues arising from generating or modeling direct treatment recommendations.
- After these adjustments, the remaining five perspectives are: *Information*, *Suggestion*, *Experience*, *Cause*, and *Question*.

We further filter the dataset to retain only well-annotated, high-quality examples. The final dataset used in our experiments consists of **3,835 question-answer threads**, split into:

- **Training set:** 2,236 threads
- **Validation set:** 959 threads
- **Test set:** 640 threads

Each thread contains a question and multiple community responses. Each response is annotated with span-level perspective labels, and each thread includes one abstractive summary per annotated perspective. This setup supports multi-label classification and multi-perspective generation, and forms the foundation for our classifier-generator architecture.

4.2 Baselines

Two baseline models were implemented:

- **Fine-tuned FlanT5 with LoRA (Low Rank Adaptation):** This model is fine-tuned using LoRA for efficient performance, significantly reducing the number of trainable parameters. It incorporates a structured prompt format and leverages an Energy-Based Loss to enforce perspective-aligned outputs.
- **GOT (GPT-OPT) with LoRA:** A custom GPT-based model trained using a few-shot learning approach. It utilizes Causal Language Modeling (CLM) loss, with masked input prompts to emphasize learning on perspective-specific content generation.

Perspective	GOT (GPT-OPT)		FlanT5 LoRA	
	BLEU	BERTScore	BLEU	BERTScore
Information	4.5425	0.8324	4.1568	0.8491
Suggestion	2.5074	0.8223	2.4202	0.8364
Experience	1.7808	0.8179	1.9625	0.8369
Question	0.4545	0.8063	0.7972	0.8366
Cause	1.7417	0.8215	2.5135	0.8544
Average	**2.2054**	**0.8201**	**2.3700**	**0.8427**

Table 2: Perspective-wise BLEU and BERTScore results for baseline models

4.3 Attention Based on Perspective:

We use the `facebook/bart-base` model as the foundational architecture for this task. The model is augmented with LoRA into the attention layers, allowing for the fine-tuning of only the LoRA adapters, while keeping the base BART model weights frozen. The LoRA configuration is set as follows:

- **Rank (r)**: 4
- **LoRA Alpha** (α): 16
- **Dropout Rate**: 0.1

The Perspective Fusion Attention mechanism projects the perspective embeddings and uses multi-head attention to fuse these embeddings with the decoder's hidden states. Specifically:

- Perspective embeddings are projected into the decoder's hidden states.
- Multi-head attention is used to combine the perspective embeddings with the decoder's hidden states in a residual manner.

The `CustomDecoderWithPerspective` class is designed to integrate this fusion mechanism. This modified decoder combines the outputs from the BART model's encoder with perspective-specific embeddings and applies the fusion mechanism to produce perspective-aligned logits.

Perspective	BLEU	BERTScore
Information	4.3210	0.8211
Suggestion	3.2007	0.8251
Experience	2.8596	0.8252
Question	0.8489	0.8206
Cause	2.9034	0.8275
Average	**2.8267**	**0.8239**

Table 3: Perspective-wise BLEU and BERTScore results for Fusion Attention model

5 Results and Analysis

We evaluate the model based on BLEU and BERTScore values for each of the five perspectives: `INFORMATION`, `SUGGESTION`, `EXPERIENCE`, `QUESTION`, and `CAUSE`. Among these, summaries generated for the `INFORMATION` perspective achieve the highest scores, with a BLEU of 10.4398 and a BERTScore of 0.8770. In contrast, the `QUESTION` perspective obtains the lowest BLEU score (0.5080), possibly due to the comparatively smaller number of training samples available for this perspective. However, its BERTScore of 0.8376 still reflects moderate semantic overlap with ground truth.

Perspective	BLEU	BERTScore
INFORMATION	10.4398	0.8770
SUGGESTION	6.4325	0.8635
EXPERIENCE	3.9180	0.8465
QUESTION	0.5080	0.8376
CAUSE	6.7420	0.8676
Average	**5.6081**	**0.8782**

Table 4: Perspective-wise BLEU and BERTScore results

BART is a strong pretrained encoder-decoder model. Adapting it with **LoRA** makes fine-tuning more efficient. Additionally, focused fine-tuning with **hard examples** enables the model to balance *fluency*, *precision*, and *semantic alignment* across all five perspectives.

BART is pretrained as both a *denoising autoencoder* and a *sequence-to-sequence model*, which

makes it particularly effective at capturing *long-range dependencies* and generating coherent summaries. **LoRA** introduces efficient parameter updates, leading to *faster convergence* and better retention of pretrained knowledge. Training on **hard examples** helps the model focus on challenging and underrepresented cases, thereby improving *robustness* and promoting a *deeper understanding of context*.

5.1 Comparative Analysis

- **Comparison with Baseline Models:**
  - **Flan-T5:**
    - Did not incorporate *perspective embeddings* or *attention cues*.
    - Treated all training samples uniformly, lacking explicit focus on perspectives.
    - Even with *LoRA*, the model failed to deeply condition generation on perspective type.
  - **GOT (GPT-OPT):**
    - Based on a *decoder-only transformer* (OPT), lacking an encoder for rich input representation.
    - Struggled to align generation with specific perspectives due to absence of explicit conditioning.
    - Generated more *generic summaries*, especially underperforming on nuanced perspectives like *Question* and *Cause*.
- **Comparison with Fusion Attention Model:**
  - Both models use *BART with LoRA* and integrate perspective information.
  - The *attention fusion model* explicitly guides attention using *perspective embeddings*.
  - The main model, however, learns better generalization by emphasizing *challenging (hard) examples* during fine-tuning.
  - Attention fusion introduces inductive bias, but may over-constrain the model or fail to capture true data complexity.
  - Hard example fine-tuning allows the model to organically learn more discriminative features from real data.
  - This led to improved scores across all perspectives, especially for *Information* and *Question*.

6 Conclusion and Future Work

6.1 Conclusion

This work presented a structured pipeline for improving abstractive summarization in healthcare question-answering tasks through progressive refinement. Initially, a classifier was employed to categorize responses into five distinct perspectives: *Information, Suggestion, Experience, Question*, and *Cause*. Subsequently, a BART-based summarization model was fine-tuned using parameter-efficient LoRA (Low-Rank Adaptation), enabling focused training with reduced resource overhead. To further enhance performance, the model underwent a second phase of fine-tuning on hard examples—responses where the initial model struggled—identified using BERTScore-based filtering. This iterative training strategy, paired with a perspective-aware design, demonstrates a robust and scalable approach to domain-specific summarization in real-world healthcare applications.

6.2 Future Work

This work demonstrates an effective classifier-generator pipeline for generating perspective-based healthcare summaries. Several avenues remain for further improvement:

- **Joint Fine-tuning of Classifier and Generator:** Although the classifier and generator are currently connected in a pipeline, jointly fine-tuning them in a multi-task setup could improve alignment between predicted perspectives and generated summaries, reducing the propagation of classification errors.
- **Improved Hard Example Mining:** The current hard-mining approach filters examples using a BERTScore threshold. Future iterations can explore contrastive learning or uncertainty-based sampling to identify more informative hard samples, potentially leading to more robust generator performance.
- **Interactive and User-Controlled Summarization:** Allowing end-users (patients or healthcare professionals) to specify desired perspectives interactively could improve usability and customization of the system.

References

1. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). *BioBERT: a*

pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

2. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., ... & Rajat, R. (2022). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv preprint arXiv:2106.09685. https://arxiv.org/abs/2106.09685

3. Zhang, X., Wang, H., Wan, X., & Liu, J. (2020). *Curriculum Learning for Natural Language Understanding*. arXiv preprint arXiv:2004.12719. https://arxiv.org/abs/2004.12719