WisPerMed @ PerAnsSumm 2025: Strong Reasoning Through Structured Prompting and Careful Answer Selection Enhances Perspective Extraction and Summarization of Healthcare Forum Threads

Tabea M. G. Pakull[1,2*], Hendrik Damm[2,3*], Henning Schäfer[1,2], Peter A. Horn[1], Christoph M. Friedrich[2,3]

[1]Institute for Transfusion Medicine, University Hospital Essen

[2]Department of Computer Science, University of Applied Sciences and Arts Dortmund

[3]Institute for Medical Informatics, Biometry and Epidemiology, University Hospital Essen

Correspondence: tabea.pakull@uk-essen.de, hendrik.damm@fh-dortmund.de

Abstract

Healthcare community question-answering (CQA) forums provide multi-perspective insights into patient experiences and medical advice. Summarizations of these threads must account for these perspectives, rather than relying on a single "best" answer. This paper presents the participation of the WisPerMed team in the PerAnsSumm shared task 2025, which consists of two sub-tasks: (A) span identification and classification, and (B) perspective-based summarization. For Task A, encoder models, decoder-based LLMs, and reasoning-focused models are evaluated under fine-tuning, instruction-tuning, and prompt-based paradigms. The experimental evaluations employing automatic metrics demonstrate that DeepSeek-R1 attains a high proportional recall (0.738) and F1-Score (0.676) in zero-shot settings, though strict boundary alignment remains challenging (F1-Score: 0.196). For Task B, filtering answers by labeling them with perspectives prior to summarization with Mistral-7B-v0.3 enhances summarization. This approach ensures that the model is trained exclusively on relevant data, while discarding non-essential information, leading to enhanced relevance (ROUGE-1: 0.452) and balanced factuality (SummaC: 0.296). The analysis uncovers two key limitations: data imbalance and hallucinations of decoder-based LLMs, with underrepresented perspectives exhibiting suboptimal performance. The WisPerMed team's approach secured the highest overall ranking in the shared task.

1 Introduction

Healthcare community question-answering (CQA) forums have become a vital resource for individuals seeking medical advice and shared experiences (Rueger et al., 2021). Unlike traditional clinical consultations, these online platforms allow users to post health-related questions and receive a wide range of answers from peers or experienced community members. Such forums often present diverse content that can address multiple aspects of a medical query. Some answers focus on personal experiences, whereas others might center on medical information or offer direct suggestions. Moreover, responses may highlight causes for a condition or pose follow-up questions to the original poster.

Despite this wealth of information, most summarization approaches for healthcare CQA threads relied on a single best-voted answer (Chowdhury and Chakraborty, 2019), which overlooks the multi-perspective nature of the discussion. A single "best" answer cannot fully encapsulate such a variety of viewpoints, highlighting the need for more perspective-aware summarization, where different types of information are distinguished rather than merged into one overarching summary.

Building on this motivation, the PerAnsSumm shared task (Agarwal et al., 2025), aims to foster research in perspective-aware healthcare answer summarization and comprises two sub-tasks:

(A) Span Identification and Classification: Given a question and user answers the task is to label spans in the answers that correspond to one of the five perspectives: *cause*, *suggestion*, *experience*, *question*, or *information*.

(B) Perspective-Based Summarization: For each perspective category, the task is to generate a concise summary that represents the content found across all answers in the thread.

This paper describes the approaches of team WisPerMed to tackle both sub-tasks. The following sections provide an overview of related work (Section 2) and describe the dataset in detail (Section 3). Then, the approaches for both tasks (Section 4) and the corresponding evaluation procedure (Section 5) are presented and their results are discussed (Section 6). Finally, the conclusion (Section 7) offers a summary of the findings.

*These authors contributed equally to this work.

Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health), pages 359–373

2 Related Work

Datasets derived from healthcare CQA forums provide insights into patient experiences (Rueger et al., 2021) and informal medical language (Chaturvedi et al., 2024). Specialized datasets (Naik et al., 2024; Chaturvedi et al., 2024; Savery et al., 2020) have been created to capture this type of content, facilitating research in patient-centered healthcare natural language processing (NLP).

Large Language Models (LLMs) demonstrate remarkable capabilities in various domains, including healthcare (Thirunavukarasu et al., 2023). BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) and its variants have formed the landscape of NLP in medicine (Thirunavukarasu et al., 2023). As encoder models, they process entire input sequences at once, leveraging attention mechanisms to build contextual representations. This ability makes them particularly well-suited for extracting structured information. Decoder-only LLMs, such as GPT (Generative Pre-trained Transformer) (Brown et al., 2020a) models, have shown impressive performance in various NLP tasks. These models process text sequentially, predicting the next token based on previous tokens. Research has explored adapting decoder-only LLMs for span labeling tasks (Dagdelen et al., 2024), leveraging their strong semantic understanding capabilities. While decoder-only LLMs excel at generating text, they face challenges in producing structured outputs. One major issue is "hallucination" (Sun et al., 2024), where models generate plausible but incorrect information. Recent advancements in LLMs have led to improved reasoning capabilities through enhanced training strategies (Pan et al., 2024) and chain-of-thought prompting (Wei et al., 2022). Models like DeepSeek-R1 (DeepSeek-AI et al., 2025) exhibit strong reasoning abilities, which are particularly valuable in healthcare applications where nuanced understanding and logical inference are crucial.

Summarization has emerged as a highly studied application of NLP in healthcare. Various approaches have been developed, including extraction- and abstraction-based techniques using LLMs (Xu et al., 2024). Perspective or aspect-based summarization (Chaturvedi et al., 2024) represents an evolving area in NLP, aiming to summarize different viewpoints or aspects within a text. This is valuable when dealing with diverse experiences and opinions expressed in online forums.

3 Dataset

The dataset used is derived from the L6 Yahoo! Answers CQA repository[1], filtered to only include health-related content. It contains 3,245 question threads with a maximum of 10 answers, totaling 10,288 individual answers. The final dataset is split into 2,236 training threads, 959 validation threads, and 50 test threads. Table 1 shows span counts, along with the number of corresponding perspective-based summaries in the training and validation sets. The raw dataset consists of a uid, user question, context to the question provided by the user, answers from other users, and raw_text which combines all information into a single string.

Perspective	Train	Val
Information	4,388 / 1,742	1,805 / 733
Cause	579 / 305	266 / 138
Suggestion	3,613 / 1,363	1,635 / 595
Question	284 / 213	131 / 101
Experience	1,245 / 745	565 / 315

Table 1: Perspective-based dataset statistics. Each cell shows the number of spans / the number of summaries.

The annotation of this dataset follows the schema described by Naik et al. (2024).

Perspective and Span Annotation. Each answer is manually reviewed to detect text spans corresponding to five perspectives: *cause*, *suggestion*, *experience*, *question*, and *information*. Annotators label these spans at the character level, conveying any of the aforementioned perspectives. As a result, a single answer can contain multiple types of perspectives. The level of granularity allows for the annotation of whitespaces and sub-words.

Perspective-Based Summarization. For each thread, a concise summary is written for every perspective observed in the answers. These summaries aim to capture the core content associated with that perspective across all answers in the thread.

4 Methods

As the sub-tasks are distinct, it is necessary to implement different approaches for each. The following sections detail the approaches employed.

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11, Last Accessed: 19. February 2025.

4.1 Task A: Span Identification and Classification

The experiments carried out for Task A used a variety of models and tuning techniques.

Models. DeBERTa-v3-large (He et al., 2021a), developed by Microsoft, builds upon the encoder model DeBERTa (He et al., 2021b). It comprises 24 layers with a hidden size of 1024, totaling approximately 418 million parameters, and is designed to enhance natural language understanding tasks. Llama-3.1-8B-Instruct was developed by Meta AI as part of the Llama series (Dubey et al., 2024) of LLMs. It contains 8 billion parameters, offering a balance between performance and computational efficiency. Llama-3.3-70B-Instruct is a 70-billion-parameter model from a newer variant of the Llama series. Both Llama models are fine-tuned with instruction-based data, enhancing their capability to follow complex directives and generate contextually relevant outputs. DeepSeek-R1 (DeepSeek-AI et al., 2025) is developed for reasoning tasks across domains such as mathematics, programming, and language. It employs a Mixture of Experts (Jacobs et al., 1991) architecture, comprising a total of 671 billion parameters. DeepSeek-R1-Distill-Llama-70B (DeepSeek-AI et al., 2025) involves distilling the DeepSeek-R1 model into a more compact form based on the Llama-3.3-70B-Instruct architecture. This involves training the smaller model (the student) to replicate the behavior of the larger DeepSeek-R1 model (the teacher) by learning from its outputs.

Fine-Tuning of Encoder Models. For the encoder approach, a DeBERTa-v3-large model was fine-tuned. The five perspective category spans were cast as NER labels in a BIO scheme (Ramshaw and Marcus, 1995; Tjong Kim Sang, 2002). During training, a maximum sequence length of 512 was set, a batch size of 16 was used, and a warmup ratio of 0.1. Model checkpoints were saved at each epoch, and the best state was chosen based on F1-Scores from the validation set. Early stopping was only applied to the *DeBERTa*$_{reconstr.\text{-}early}$ model. For inference, the raw_text style representation was available for the training and validation data only, but not for the test set. Therefore, two inference approaches were explored. *DeBERTa*: Each individual answer was provided to the model as a separate input, and the resulting token-level predictions were stored on a per-answer basis. *DeBERTa*$_{reconstr.}$: Each test sample was reconstructed into a single sequence by inserting the same markers (`uri: <ID>`, `question: <text>`, and `answer_0: <text>`) to obtain a format that is consistent with the training data. The entire thread was then passed to the model at once, enabling it to capture cross-answer context. After token-level predictions were generated for both approaches, a chunk-merging step was applied to merge consecutive tokens that shared the same perspective class. Single-word spans were removed to improve precision. The final labeled segments were then saved in the submission format.

Instruction-Tuning of Llama-3.1-8B-Instruct. In order to optimize Llama-3.1-8B-Instruct for perspective-aware span extraction, the train split of the dataset was structured into a format suitable for instruction-tuning (Wei et al., 2021). Instruction-tuning refers to the process of training LLMs on data formatted as instructions. Input and output are transformed in a conversation-style format containing a system and user prompt as well as the structured assistant output. In this work the system prompt outlines the task, classification guidelines, and output format. To ensure the consistency and successful parsing of outputs, the model is instructed to return its response as a TypeScript object. The user prompt contains the answers from forum users and the assistant output contains the spans structured as a TypeScript object. All prompts can be found in the Appendix A.5.1.

To maintain computational efficiency Parameter-Efficient Fine-Tuning (PEFT) (Ding et al., 2023) via LoRA (Low-Rank Adaptation) (Hu et al., 2022) was employed. More details on the implementation can be found in the Appendix A.2.

During inference, the instruction-tuned model utilizes the same prompts as in training. The inference parameter are available in Appendix A.3.1.

Prompt-Based Techniques. To complement fine- and instruction-tuning, zero-shot and few-shot prompting strategies (Brown et al., 2020b) were employed. These methods instruct LLMs to extract relevant spans and classify them into the correct perspective category without the need for parameter updates.

In the zero-shot setting, the model is directly prompted using the system prompt that outlines the task, classification guidelines, and output format, combined with the user prompt that contains the answers from forum users. This method tests

the model's ability to generalize its understanding of text span classification based solely on its pre-trained knowledge.

To enhance performance, few-shot learning was introduced by showing the model examples of gold-standard output in a conversational style. These examples demonstrate how the spans should be extracted and categorized, helping the model learn through analogy. Two variations of few-shot prompting were explored: Standard few-shot prompting, where gold-standard examples were provided as part of the same interaction and few-shot prompting with system message resets, where each example was treated as an independent instance with repeated system prompts to reinforce adherence to the task and the output format.

In both few-shot and zero-shot settings the same system and user prompts are used as for instruction-tuning (see Appendix A.5.1).

4.2 Task B: Perspective-Based Summarization

Early experiments on the validation set indicated that fine-tuning models solely with span data for the summarization task led to suboptimal results. Relying solely on span annotations failed to capture the broader contextual and query-specific nuances necessary for generating high-quality summaries. Furthermore, when using spans as input, performance on Task B is dependent on Task A performance. Consequently, a more comprehensive instruct-tuning strategy was adopted that leverages all available information, including the context, question, and answers. In this revised approach, models are exposed to a richer set of inputs during the training process, enabling improved understanding and synthesis of relevant information for summarization. The instruct-tuning was tested on the following four models. The prompts for the instruction-tuning can be seen in Appendix A.5.2.

Models. Mistral-Small-24B-Instruct[2] is a pre-trained, instruction-tuned model that achieves performance comparable to larger models such as Llama 3.3 70B while offering faster inference. Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). BioMistral-7B-DARE (Labrak et al., 2024) adapts Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) for the biomedical domain through additional pre-training on PubMed Central, achieving strong results on medical question-answering benchmarks and effective multilingual generalization. DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI et al., 2025) is a distilled dense model that replicates the reasoning patterns of the larger DeepSeek-R1 (DeepSeek-AI et al., 2025) in a compact form.

[2] https://mistral.ai/news/mistral-small-3, Last Accessed: 23. February 2025.

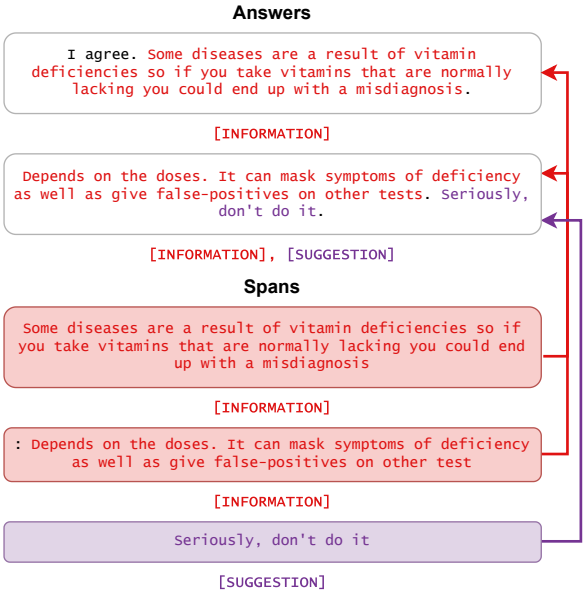

Figure 1: Workflow Diagram of the Answer Labeling Pipeline for Task B pre-classification. The process begins by extracting answer boundaries from raw_text. Next, labeled spans are assigned to their corresponding answers based on their starting index. Finally, the original answer texts are assigned the perspective labels of contained spans.

4.2.1 Pre-classification Methodology

Instead of using all answers to generate a summary for a given perspective, multi-label perspective classifiers were trained using DeBERTaV3 and Mistral-7B-v0.3. To create a labeled answer dataset, answer spans were extracted and the corresponding answers determined via regular expressions (see Figure 1). In some instances, a more complex regex was needed to fix annotation errors; for example, the second span in Figure 1 mistakenly included a leading colon and whitespace from raw_text that were not present in the original answer.

The trained classifiers were then applied to the test set to label answers and generate summaries, as illustrated in Figure 2. For model instruct-tuning, only answers labeled with the same perspective as the requested summary (e.g., *information*) were used. If no answers were labeled with the desired perspective, the model used all available answers instead. This strategy ensures that every thread receives one summary per perspective, regardless

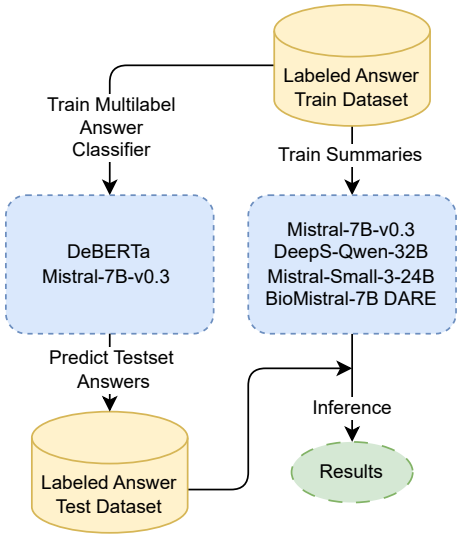

Figure 2: The labeled answer train dataset was used to train multi-label classifiers and instruction-tune models for Task B. The test dataset, with predicted answer perspectives, was then used to generate summarizations.

of the distribution of labeled answers. Additionally, an alternative approach involves training five separate models (Mistral-7B-v0.3_{5x}), one for each perspective.

5 Evaluation

A range of evaluation metrics are used to evaluate different aspects of the results, with scores in Table 3 and Table 6 provided by the shared task organizers (Agarwal et al., 2025).

5.1 Task A: Span Identification and Classification

The evaluation methodology for Task A comprises assessment of classification performance and span identification accuracy. The former is measured using a macro-averaged F1-Score (Macro F1) and a weighted F1-Score (Weight F1). The latter is evaluated using Strict and Proportional Matching (Agarwal et al., 2025). Strict Matching involves the evaluation of the exact match between predicted and gold standard spans, with precision (P), recall (R), and F1-Scores being computed from the number of exact matches. Proportional Matching allows for partial correctness by evaluating the token-level overlap between predicted and gold-standard spans. The number of overlapping tokens in each predicted span is measured against the most similar gold span, and the results are then used to compute precision, recall, and F1-Scores. This approach makes it more flexible than strict matching.

To evaluate hallucinated spans in LLM-generated outputs, it is checked whether the output spans appear verbatim in the original answers. This analysis reports the proportion of correctly extracted spans, providing a quantitative measure of the model's tendency to introduce extraneous content. This analysis is reported in this work in addition to the shared task results, and is not used for ranking.

5.2 Task B: Perspective-Based Summarization

In Task B the evaluation methodology employs multiple automatic metrics to assess the quality of generated summaries across the aspects relevance and factuality.

5.2.1 Relevance

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) measures the F1-Score of overlap of unigrams (ROUGE-1), bigrams (ROUGE-2), and longest common subsequences (ROUGE-L) between the generated and reference summaries. Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) is a metric that evaluates the precision of n-gram overlap. Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee and Lavie, 2005) considers both synonymy and stemming to provide a more flexible assessment of lexical similarity. It also calculates the degree to which the matched words are ordered in the same way in the summary as in the reference. BERTScore (Zhang et al., 2020) leverages contextualized embeddings from BERT to compute semantic similarity between generated and reference summaries.

5.2.2 Factuality

The AlignScore (Kryscinski et al., 2020) quantifies the degree of alignment between the facts in the summary and the reference. SummaC-Conv (Laban et al., 2022) (SummaC) detects inconsistencies by segmenting documents into sentence-level pairs and using a convolutional layer to aggregate entailment scores for the factuality assessment.

6 Results and Discussion

The final rankings of the top five participating teams in the shared task are summarized in Table 2. The WisPerMed team achieved the highest overall ranking (0.457) in the shared task, narrowly outperforming the other teams. The ranking is based on both sub-tasks. In Task A WisPerMed obtained a

		Ovr.	Task A	Task B	
#	Team	$\bar{x}$	$\bar{x}$	Rel.	Fact.
1	WisPerMed	**0.457**	0.598	0.421	0.352
2	YALENLP	0.455	0.604	**0.436**	0.325
3	yxyx	0.453	**0.621**	0.365	**0.372**
4	AICOE	0.45	0.605	0.395	0.348
5	KHU_LDI	0.449	0.589	0.417	0.343

Table 2: Final results of the top five teams in the shared task. Columns show team rank (#) and average scores ($\bar{x}$) for overall (Ovr.) and Task A. Task B scores are reported separately for relevance (Rel.) and factuality (Fact.). **Bold** values indicate the highest score, and underlined values mark the second-highest.

score of 0.598 using DeepSeek-R1 in the zero-shot setting (DeepS-R1$_{zs}$ in Table 3). Task B is further divided into relevance and factuality categories, where WisPerMed ranked first in both categories combined using the instruction-tuned Mistral-7B-v0.3, with the labeled answer test dataset (Mistral-7B-v0.3$_{pre\text{-}class.}$).

6.1 Task A: Span Identification and Classification

Table 3 summarizes the performance of various experimental setups for Task A. Evaluation metrics include Macro F1-Score, Weighted F1-Score, and precision (P), recall (R), and F1-Scores under both Strict and Proportional span matching.

DeepSeek-R1 in the zero-shot setting (DeepS-R1$_{zs}$) achieved the best scores in Macro F1-Score (0.878), Weighted F1-Score (0.921), and several span matching metrics (Strict Recall (0.229), Strict F1-Score (0.196), Proportional Recall (0.738), and Proportional F1-Score (0.676)). Its high recall values under both matching criteria indicate robust retrieval capabilities. Moreover, its overall average score of 0.598 reinforces its superior performance across the evaluation metrics.

DeBERTa achieves an overall score of 0.539, yet it does not exhibit any particular advantage in individual sub-metrics. The DeBERTa-based variants DeBERTa$_{reconstr.\text{-}early}$ and DeBERTa$_{reconstr.}$ exhibit improved performance. The former attained the second-best Macro F1-Score (0.875), while the latter secured the second-best Weighted F1-Score (0.909) and the highest Proportional Precision (0.627). This observation indicates that smaller transformer-based models, specifically optimized for sequence labeling tasks, can demonstrate comparable performance to larger general-purpose LLMs in perspective-aware span extraction, despite their smaller size. Making them a considerable choice to reduce resource cost (computational and environmental).

The Llama-based models show a clear dependence on model size and training paradigm. The instruction-tuned Llama-3.1-8B-Instruct (Llama-3.1-8B$_{it}$) underperforms, with a Macro F1-Score of 0.602 and a Strict F1-Score of 0.023, indicating the limitations of smaller decoder-only models for this task. This performance discrepancy could also indicate that the instruction-tuning process was not sufficiently rigorous or tailored for this specific task. In contrast, the larger Llama-3.3-70B-Instruct variants show enhanced performance. Llama-3.3-70B$_{fs\text{-}sys.}$ variant achieved the highest Strict Precision (0.182) as well as competitive Strict Recall (0.192) and Strict F1-Score (0.187), suggesting that repeated system message enhance the model's ability to precisely identify spans. Its overall average performance of 0.580 places Llama-3.3-70B$_{fs\text{-}sys}$ in second place among WisPerMed's approaches.

The enhanced reasoning capabilities in DeepS-R1 and it's much larger size might have contributed to its superior overall performance. The notable improved overall score of the distilled version (DeepS-Llama-3.3-70B$_{fs}$) compared to the original Llama-3.3-70B-Instruct (Llama-3.3-70B$_{fs}$) in the few-shot setting underscore this hypothesis about the impact of reasoning on span labeling performance.

All models exhibited lower scores under Strict span matching, with the highest Strict F1-Score reaching only 0.196. This consistent difference indicates that precise boundary prediction remains a difficult aspect of span extraction. This may be attributed to boundary misalignments in span extraction, where models correctly identify relevant content but fail to precisely match the annotated span boundaries. It may also stem from inconsistencies in the annotated dataset (see Section 4.2.1), where spans include partial words, trailing or preceding whitespaces. The DeepS-R1$_{zs}$ model's superior performance in Strict metrics confirms its ability to accurately retrieve relevant spans, even under exacting conditions. Proportional F1-Scores ranged from 0.420 (Llama-3.1-8B$_{it}$) to 0.676 (DeepS-R1$_{zs}$). The overall higher scores for proportional matching suggests that many of the errors in strict matching are due to minor boundary misalignments rather than completely incorrect span predictions. Even with the best approaches among the top five teams in the shared task, performance remains sub-optimal, underscoring the inherent complexity and

Experiment	Macro F1	Weight F1	Str. P	Str. R	Str. F1	Prop. P	Prop. R	Prop. F1	$\bar{x}$
DeBERTa	0.855	0.906	0.103	0.126	0.113	0.600	0.593	0.596	0.539
DeBERTa$_{\text{reconstr.-early}}$	0.875	0.907	0.170	0.152	0.161	0.619	0.621	0.620	0.563
DeBERTa$_{\text{reconstr.}}$	0.871	0.909	0.115	0.116	0.115	**0.627**	0.584	0.605	0.543
Llama-3.1-8B$_{\text{it}}$	0.602	0.733	0.028	0.019	0.023	0.319	0.616	0.420	0.392
Llama-3.3-70B$_{\text{fs}}$	0.828	0.887	0.065	0.048	0.055	0.561	0.604	0.582	0.508
Llama-3.3-70B$_{\text{fs-sys.}}$	0.866	0.907	**0.182**	0.192	0.187	0.606	0.689	0.645	0.580
DeepS-Llama-70B$_{\text{fs}}$	0.839	0.882	0.174	0.162	0.168	0.516	0.647	0.574	0.541
DeepS-R1$_{\text{zs}}$	**0.878**	**0.921**	0.171	**0.229**	**0.196**	0.623	**0.738**	**0.676**	**0.598**

Table 3: Results for Task A. Columns show Macro F1-Score (Macro F1) and Weighted F1-Score (Weight F1), along with precision (P), recall (R), and F1-Scores under Strict (Str.) and Proportional (Prop.) span matching for all experiments. The final column ($\bar{x}$) represents the overall average score. The best values are highlighted in **bold**, while the second-best values are underlined. Abbreviations: it - instruction-tuned, fs - few-shot, fs-sys. - few-shot with repeated system messages, zs - zero-shot.

Experiment	Found Spans (%)
Llama-3.1-8B$_{\text{it}}$	90.70
Llama-3.3-70B$_{\text{fs}}$	96.60
Llama-3.3-70B$_{\text{fs-sys.}}$	97.65
DeepS-Llama-3.3-70B$_{\text{fs}}$	80.82
DeepS-R1$_{\text{zs}}$	92.02

Table 4: Percentage of generated spans that match verbatim spans in the original answers. Abbreviations: it - instruction-tuned, fs - few-shot, fs-sys. - few-shot with repeated system messages, zs - zero-shot.

challenges of perspective-based span labeling.

In addition to the shared task evaluation metrics, an analysis was conducted to quantify hallucinated content in LLM-generated outputs (see Table 4). For instance, Llama-3.1-8B$_{\text{it}}$ achieved an overall percentage of 90.70%, indicating that a notable fraction of its output spans deviated from the source text. In contrast, the Llama-3.3-70B variants exhibited a higher match percentage of 96.60% and 97.65%, suggesting improved fidelity to the input text. However, the DeepSeek-R1-Distill-Llama-70B variant showed a considerably lower match percentage (80.82%), underscoring a higher tendency to generate hallucinated or extraneous spans. The DeepS-R1$_{\text{zs}}$ model yielded 92.02%, indicating, that reasoning may lead to a higher tendency to introduce extraneous content.

6.2 Task B: Perspective-Based Summarization

Results of Task B are discussed using metrics for factuality (AlignScore and SummaC) and relevancy (ROUGE, BERTScore, METEOR, and BLEU).

Answer Pre-classification Table 5 presents the classification performance of trained Mistral-7B-v0.3 and DeBERTaV3 on the validation set. Mistral-7B-v0.3 achieves a higher Macro F1-Score (0.577) compared to DeBERTaV3 (0.510). Both models perform well on perspectives with ample training data, such as *experience* and *suggestion*. However, the *cause* perspective, which has limited training examples, shows a very low F1-Score of 0.030 for DeBERTaV3. This contrast reveals the impact of training data scarcity on classification performance. Overall, while both models effectively classify well-represented perspectives, Mistral-7B-v0.3 exhibits a more balanced performance across classes, highlighting the challenge of underrepresentation in certain categories. Therefore Mistral-7B-v0.3 was chosen to classify the test dataset answers.

Perspective	P	R	F1	S
Mistral-7B-v0.3				
experience	0.735	0.683	0.708	419
suggestion	0.718	0.749	0.733	1,142
cause	0.571	0.124	0.204	193
question	0.851	0.381	0.526	105
information	0.704	0.722	0.713	1,210
Macro	**0.716**	**0.532**	**0.577**	3,069
Weighted	**0.710**	0.677	**0.681**	3,069
DeBERTaV3				
experience	0.671	0.780	0.722	419
suggestion	0.732	0.762	0.746	1,142
cause	0.300	0.016	0.030	193
question	0.778	0.200	0.318	105
information	0.689	0.786	0.734	1,210
Macro	0.634	0.509	0.510	3,069
Weighted	0.681	**0.708**	0.679	3,069

Table 5: Comparison of classification performance on the validation set for Mistral-7B-v0.3 and DeBERTa. In the overall Macro and Weighted rows, the best score (between models) for each metric is shown in **bold**. Precision (P), recall (R), F1-Score (F1), and Support (S) are reported.

Experiment	R1	R2	RL	BERT	MET	BLEU	Rel.	Align	SC	Fact.
BioMistral-7B	0.344	0.151	0.308	0.753	0.286	0.108	0.325	**0.449**	0.276	0.363
Mistral-7B-v0.3 1E	0.408	0.182	0.371	0.891	0.378	0.091	0.387	0.369	0.260	0.314
Mistral-7B-v0.3$_{5x}$	0.445	**0.222**	0.406	**0.899**	0.406	0.127	0.418	0.421	**0.306**	0.364
Mistral-7B-v0.3$_{\text{pre-class.}}$ 1E	0.437	0.211	0.397	0.897	0.397	0.123	0.410	0.441	0.297	**0.369**
Mistral-7B-v0.3$_{\text{pre-class.}}$ 2E	**0.452**	0.221	**0.410**	**0.899**	**0.410**	**0.135**	**0.421**	0.409	0.296	0.352
Mistral-Small-3-24B	0.291	0.088	0.255	0.877	0.251	0.048	0.302	0.393	0.238	0.316
DeepS-Qwen-32B	0.299	0.097	0.264	0.862	0.249	0.067	0.306	0.372	0.241	0.306

Table 6: Results for Task B. This table reports ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL), BERTScore (BERT), METEOR (MET), BLEU, Relevance average (Rel.), AlignScore (Align), SummaC-Conv (SC), and Factuality average (Fact.). The best values are highlighted in **bold**, while the second-best values are underlined. Abbreviations: pre-class. - pre-classified answers, E - epoch.

Labeled Test Dataset The test dataset consists of 231 answers in total. Among these, the predicted perspectives are distributed as follows: 85 answers were labeled as *experience*, 112 as *suggestion*, 15 as *cause*, 12 as *question*, and 93 as *information*. This distribution mirrors the one in the validation set. Labels such as *suggestion* and *information* are common, while the *cause* and *question* perspectives are notably underrepresented. This suggests that the prediction of answers is robust and the proportions of predicted labels are consistent with expectations. The threshold of the classifier for each perspective was determined by using the validation set. Detailed information on the classifier (F1-Score, P, R) can be found in Appendix A.4.

Summarization Results The results in Table 6 illustrate the performance of various models on Task B. Notably, Mistral-7B-Instruct-v0.3 with pre-classification (Mistral-7B-v0.3$_{\text{pre-class.}}$) trained for two epochs achieved the best overall performance, with the highest ROUGE-1 (0.452) and ROUGE-L (0.410) scores, as well as top scores in BERTScore (0.899), METEOR (0.410), and BLEU (0.135). This indicates that the approach of pre-classifying answers prior to instruct-tuning notably enhanced the quality of the generated summaries by improving relevance. The five-model approach (Mistral-7B-v0.3$_{5x}$), where a separate model was trained for each perspective, also performed very well. It ranks first in ROUGE-2 (0.222) and SummaC (0.306) and second in multiple other metrics. In contrast, Mistral-Small-24B-Instruct (Mistral-Small-3-25B) and the distilled Qwen model (DeepS-Qwen-32B) yielded lower scores, while BioMistral-7B performed moderately but did not match the performance of the pre-classification approaches. Furthermore, the relevancy and factuality averages provide additional insight. Higher relevancy scores suggest that the summaries are closely aligned with the intended perspectives, and better factuality scores indicate fewer factual errors. In particular, the pre-classification approach achieved a robust relevancy average (0.421) and acceptable factuality (0.352), underscoring its ability to capture and synthesize perspective-specific content effectively. Overall, these findings confirm that integrating an answer pre-classification step leads to superior summarization performance, making it the best overall strategy for Task B.

7 Conclusion

In conclusion, the study presents an investigation into perspective-aware summarization for healthcare CQA forums through two interrelated tasks: (A) span identification and classification, and (B) perspective-based summarization. The experimental results demonstrate that while fine-tuned encoder models such as DeBERTaV3 yield competitive performance in precise span extraction, the integration of enhanced reasoning capabilities, as seen in DeepSeek-R1, leads to superior overall performance in capturing complex contextual cues. The analysis of hallucinated content reveals that model fidelity to the source text remains a critical challenge, particularly for larger decoder-only models employing reasoning mechanisms. The findings from the summarization experiments underscore the efficacy of an answer pre-classification strategy, which improves both relevancy and factuality of generated summaries by effectively leveraging perspective-specific information.

Limitations

This work has several limitations that should be addressed in future research.

One limitation is the data imbalance inherent in the dataset. The underrepresentation of certain

classes in the dataset negatively impacts the classifier's performance as well as robustness of the evaluation. It highlights a broader challenge in obtaining balanced annotations in perspective-based datasets.

Another limitation concerns the generation of summaries for each perspective regardless of the presence of corresponding spans. Since there was no penalty for generating summaries for perspectives without supporting evidence, the system produced what may be considered "useless" summaries. Future evaluations should consider incorporating a penalty for such outputs to better reflect the accuracy and utility of the generated summaries.

Automatic evaluation metrics may not capture all aspects of healthcare summarization, such as clinical relevance and interpretability, potentially leading to an incomplete assessment of model performance.

Acknowledgments

The work of Henning Schäfer, Hendrik Damm, and Tabea M. G. Pakull was funded by a PhD grant from the DFG Research Training Group 2535 *Knowledge- and data-based personalisation of medicine at the point of care (WisPerMed)*.

References

Siddhant Agarwal, Md Shad Akhtar, and Shweta Yadav. 2025. Overview of the peranssumm 2025 shared task on perspective-aware healthcare answer summarization. In *Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health) @ NAACL 2025*, Albuquerque, USA. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Rochana Chaturvedi, Abari Bhattacharya, and Shweta Yadav. 2024. Aspect-oriented consumer health answer summarization. *Preprint*, arXiv:2405.06295.

Tanya Chowdhury and Tanmoy Chakraborty. 2019. CQASUMM: Building References for Community Question Answering Summarization Corpora. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, CODS-COMAD '19, pages 18–26, New York, NY, USA. Association for Computing Machinery.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. DeBERTaV3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. BioMistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864, Bangkok, Thailand. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, pages 74–81, Barcelona, Spain.

Gauri Naik, Sharad Chandakacherla, Shweta Yadav, and Md Shad Akhtar. 2024. No perspective, no perception!! perspective-aware healthcare answer summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15919–15932, Bangkok, Thailand. Association for Computational Linguistics.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically Correcting Large Language Models: Surveying the Landscape of Diverse Automated Correction Strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*.

Jasmina Rueger, Wilfred Dolfsma, and Rick Aalbers. 2021. Perception of peer advice in online health communities: Access to lay expertise. *Social Science & Medicine*, 277:113117.

Max Savery, Asma Ben Abacha, Soumya Gayen, and Dina Demner-Fushman. 2020. Question-driven summarization of answers to consumer health questions. *Scientific Data*, 7(1):322.

Yujie Sun, Dongfang Sheng, Zihan Zhou, and Yifei Wu. 2024. AI hallucination: Towards a comprehensive classification of distorted information in artificial intelligence-generated content. *Humanities and Social Sciences Communications*, 11(1):1–14.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 24824–24837, Red Hook, NY, USA. Curran Associates Inc.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: a survey. *Frontiers of Computer Science*, 18(6):186357.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Appendix

The appendix provides additional details on the frameworks and models used in this work, including their licensing terms, the setup for instruction-tuning, decoding parameters, and the specific prompting strategies employed in the experiments.

A.1 Licences

The frameworks and models used in this work are governed by different open-source licenses, as detailed in Table 7.

Framework/Model	License
unsloth[3]	Apache-2.0
deberta-v3-large[4]	MIT
Llama-3.1-8B-Instruct[5]	Llama 3.1 Comm.
Llama-3.3-70B-Instruct[6]	Llama 3.3 Comm.
DeepSeek-R1-Distill-Llama-70B[7]	MIT
DeepSeek-R1[8]	MIT
Mistral-7B-Instruct-v0.3[9]	Apache-2.0
Mistral-Small-24B-Instruct-2501[10]	Apache-2.0
BioMistral-7B-DARE[11]	Apache-2.0
DeepSeek-R1-Distill-Qwen-32B[12]	MIT

Table 7: Licensing terms for each framework and model used in this work, including various Apache-2.0 (Apache License 2.0), MIT (Massachusetts Institute of Technology License), and Comm. (Llama Community License).

A.2 Training Setup

This section outlines the configurations, including parameter-efficient tuning, and optimization methods used for training the models.

A.2.1 Task A: Span Identification and Classification

For the span identification and classification task, the Llama-3.1-8B-Instruct model was fine-tuned using PEFT techniques via LoRA. The unsloth framework was used to optimize training. The training examples were structured as shown in Figure 3. The training utilized AdamW 8-bit optimization, with a learning rate of $2e-5$, batch size of 1, and gradient accumulation steps of 64. The model was trained for two epochs.

A.2.2 Task B: Perspective-Based Summarization

The instruction-tuning parameters can be seen in Table 8.

The answer classifier based on Mistral-7B-v0.3 was trained for two epochs using FP16, with a batch size of 1 and gradient accumulation over 4 steps. It employed a learning rate of $2e$-5, a maximum sequence length of 4096, and utilized LoRA with a rank of 8 and an alpha of 16. In contrast, the DeBERTaV3-base model was trained for two epochs with a learning rate of $2e$-5, a batch size of 16, and a maximum sequence length of 1024.

[3] https://unsloth.ai/, Last Accessed: 24 February 2025.

[4] https://huggingface.co/microsoft/deberta-v3-large, Last Accessed: 24. February 2025.

[5] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, Last Accessed: 24. February 2025.

[6] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct, Last Accessed: 24. February 2025.

[7] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B, Last Accessed: 24. February 2025.

[8] https://huggingface.co/deepseek-ai/DeepSeek-R1, Last Accessed: 24. February 2025.

[9] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3, Last Accessed: 24. February 2025.

[10] https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501, Last Accessed: 24. February 2025.

[11] https://huggingface.co/BioMistral/BioMistral-7B-DARE, Last Accessed: 24. February 2025.

[12] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B, Last Accessed: 24. February 2025.

Model	E	FP16	lr	Batch	GA	LR	LA	DO	MSL	TM
Mistral-7B-v0.3	1/2	True	$2e$-5	6	–	16	32	0.1	4096	All linear
Mistral-Small-24B-Instruct-2501	2	True	$2e$-5	1	4	4	8	0.1	1400	All linear
DeepSeek-R1-Distill-Qwen-32B	2	True	$2e$-5	1	–	8	16	0.1	2024	All linear
BioMistral-7B-DARE	2	True	$2e$-4	6	–	32	64	0.1	4096	All linear

Table 8: Instruction-tuning hyperparameters for the models. Abbreviations: E - epochs; FP16 - FP16 training; lr - learning rate; Batch - batch size; GA - gradient accumulation steps; LR - LoRA rank; LA - LoRA alpha; DO - dropout; MSL - maximum sequence length; TM - targeted modules. Note that Biomistral Dare and Mistral v03 instruct share the same hyperparameters as Mistral-7B-v0.3.

A.3 Decoding Setup

This section outlines the inference procedures used to generate spans and summarization.

A.3.1 Task A: Span Identification and Classification

Llama-3.3-70B-Instruct was deployed using vLLM (Kwon et al., 2023), an inference framework designed for efficient text generation. The model was accessed via the OpenAI python package[13] version 1.60.0, with default sampling parameters[14] except for `max_tokens`, which was set to 2000. For the instruction-tuned model Llama-3.1-8B-Instruct, inference was performed using the unsloth framework. Outputs were generated with default configuration[15] but `max_new_tokens` set to 5000 and a 1.2 repetition penalty.

A.3.2 Task B: Perspective-Based Summarization

For inference, all models were configured with a maximum sequence length of 4.096 tokens, up to 1.024 new tokens, deterministic decoding (do sample set to false), and a temperature of 0.7. The only exception is DeepSeek-R1-Distill-Qwen-32B, which was run with a temperature of 0.6 while all other inference parameters remained the same.

A.4 Classifier

Table 9 details the threshold tuning experiments for the Mistral-7B-v0.3 model on the validation set. For each perspective, the optimal threshold is reported alongside the corresponding precision, recall, and F1-Scores for both class 0 and class 1.

[13] https://github.com/openai/openai-python, Last Accessed: 24. February 2025.

[14] https://docs.vllm.ai/en/latest/api/inference_params.html, Last Accessed: 24. February 2025.

[15] https://huggingface.co/docs/transformers/v4.49.0/en/main_classes/text_generation#transformers.GenerationConfig, Last Accessed: 24. February 2025.

Perspective	T	Class	P	R	F1
experience	0.25	0	0.971	0.945	0.958
		1	0.700	0.819	0.755
suggestion	0.25	0	0.903	0.747	0.818
		1	0.666	0.863	0.751
cause	0.15	0	0.971	0.936	0.953
		1	0.375	0.580	0.455
question	0.15	0	0.990	0.988	0.989
		1	0.673	0.705	0.688
information	0.40	0	0.849	0.752	0.797
		1	0.671	0.790	0.726

Table 9: Threshold (T) tuning results on the validation set for the Mistral-7B-v0.3 model. For each perspective, the optimal threshold and the corresponding precision, recall, and F1-Scores for class 0 and class 1 are reported.

For instance, for the *experience* perspective, a threshold of 0.25 yields excellent performance for class 0 (P = 0.971, R = 0.945, F1 = 0.958) and solid results for class 1 (P = 0.700, R = 0.819, F1 = 0.755). In contrast, the *cause* perspective exhibits a very low F1-Score of 0.455 for class 1 despite high performance for class 0. These results demonstrate that while well-supported classes achieve high scores, those with fewer examples remain difficult to classify accurately.

A.5 Prompting

This section details the design of system and user prompts, including formatting strategies for both sub-tasks.

A.5.1 Task A: Span Identification and Classification

The prompting setup is designed to ensure structured, consistent, and accurate extraction of perspective-based spans. The motivation behind this approach was to align the model's pretraining with the task requirements, leveraging the instruction-following capabilities of LLMs that have undergone instruction-tuning. Since such

System:

You are an advanced AI model specializing in perspective-aware span extraction. Your objective is to analyze health-related community question-answering forums, where users ask health-related questions and receive multiple answers containing different perspectives.
Task
Identify relevant spans (text segments) within user-provided answers that correspond to one or more of the five perspective categories:
CAUSE: It underlines the potential cause of a medical phenomenon or a symptom. It answers the "Why" regarding a specific observation, offering insights to identify the root cause.
SUGGESTION: It encapsulates strategies, recommendations, or potential courses of action towards management or resolution of a health condition.
EXPERIENCE: It covers first-hand experiences, observations, insights, or opinions derived from treatment or medication related to a particular problem.
QUESTION: It consists of interrogative phrases, follow-up questions and rhetorical questions that are sought to better understand the context. They typically start with phrases like Why, What, Do, How, and Did etc, and end in a question mark.
INFORMATION: It encompasses segments that offer factual knowledge or information considering the given query. These segments provide comprehensive details on diagnoses, symptoms, or general information on a medical condition.
Classify each identified span into the correct perspective category based on its meaning and intent.
Guidelines for Identifying and Classifying Spans:
Select complete spans. Avoid excessively short spans that lack context.
Only include spans that align with a perspective category.
Never change the wording or formatting of the spans. EXTRACT and not rewrite.
Output Format
Your response must always be one valid PerspectiveSpans object:

```typescript
interface PerspectiveSpans {
   EXPERIENCE: string[],
   INFORMATION: string[],
   CAUSE: string[],
   SUGGESTION: string[],
   QUESTION: string[]
}
```

Each perspective category should contain a list of spans extracted from the answers. If no span belongs to a category, leave an empty list. Do not add additional perspectives.
Example
User Input:
"answers": [
<answer$_1$>,
...
<answer$_n$>
]
Assistant output

```typescript
const spans: PerspectiveSpans = {
   "CAUSE": [<cause_span$_1$>, <cause_span$_2$>],
   "SUGGESTION": [<suggestion_span$_1$>, ..., <suggestion_span$_n$>],
   "EXPERIENCE": [],
   "QUESTION":[],
   "INFORMATION": [<information_span$_1$>]
}
```

Figure 3: System prompt for Task A defining the task, perspective categories, and extraction guidelines for structured span identification.

models are trained to interpret and execute user instructions, framing the task in a conversational format was a natural way to enhance compliance with task constraints.

The system prompt (see Figure 3) was designed to provide precise definitions and distinguishing criteria for each of the five perspectives. These explicit definitions help the model differentiate between similar categories and prevent incorrect or overly broad span selections. Furthermore, the system prompt reinforces extraction constraints, ensuring that the model preserves the wording and formatting of the original text in the user prompt (see Figure 4) rather than generating new or para-

User:

"answers": [
<answer$_1$>,
...
<answer$_n$>
]

Figure 4: User prompt for Task A providing the input format with a list of answers from a the discussion thread.

Assistant:

```typescript
const spans: PerspectiveSpans = {
   "CAUSE": [<cause_span$_1$>, <cause_span$_2$>],
   "SUGGESTION": [<suggestion_span$_1$>, ..., <suggestion_span$_n$>],
   "EXPERIENCE": [],
   "QUESTION":[],
   "INFORMATION": [<information_span$_1$>]
}
```

Figure 5: Assistant response for Task A demonstrating the structured TypeScript-like format for extracted spans.

phrased spans. The example used in the system prompt to demonstrate the formats is the training example with uri 1504599.

Another critical consideration was the need for structured outputs to facilitate automatic evaluation. Since LLMs generate open-ended text by default, responses can vary notably in format if not explicitly constrained. To address this, the output structure was formatted as a TypeScript-like object (see Figure 5), enforcing a predefined schema where extracted spans are categorized under their respective perspective labels.

Beyond instruction-tuned training, the prompting framework was also applied to zero-shot and few-shot inference settings to assess the model's ability to generalize its span extraction capabilities without direct fine-tuning. The zero-shot setting tested whether the model could infer the extraction rules solely from the system prompt, while the few-shot setting provided additional gold-standard examples. In the standard few-shot setting, multiple examples were included in the same conversation, allowing the model to learn span extraction through analogy. In contrast, the few-shot with repeated system message reinforced consistency by repeating the system prompt before each example.

A.5.2 Task B: Perspective-Based Summarization

Figure 6 shows the prompt for the instruction-tuning of the summarization task, on the example of the *information* perspective. For other perspectives, only the "Perspective Instruction" was changed:

- **Information Perspective:** For information purposes, generate a concise summary.
- **Suggestion Perspective:** It is suggested, generate a concise summary with suggestions.
- **Experience Perspective:** One user shared his experience, generate a concise summary.
- **Cause Perspective:** Some of the causes, generate a concise summary.
- **Question Perspective:** It is inquired, generate a concise summary addressing the questioner.

System:

Task: In recent years, healthcare community question-answering (CQA) forums have allowed users to seek advice and share experiences. Answers may contain diverse perspectives, such as factual information, suggestions, personal experiences, potential causes, or follow-up questions. The goal is to generate a concise, perspective-specific summary that accurately reflects the essence of the relevant content from the discussion.

Writing Guidelines:

1. Analyze the provided information to capture essential ideas and significant medical details.
2. Generate a concise summary that reflects the core essence of the annotated perspective.
3. Frame the summary as follows:
   - For Information summaries, begin with phrases like "For information purposes."
   - For Suggestion summaries, begin with phrases such as "It is suggested," "It is advised," or "Consider."
   - For Experience summaries, begin with phrases like "One user shared his experience" or "In user's experience."
   - For Cause summaries, begin with phrases like "Some of the causes."
   - For Question summaries, begin with phrases such as "It is inquired."
4. Do not add any information beyond what is provided.

Perspective Instruction: {e.g., "For information purposes, generate a concise summary."}

Question: <QUESTION>

Context: <CONTEXT>

Answers:
- <answer$_1$>
- <...>
- <answer$_n$>

Information Summary:

User:

<Summary>

Figure 6: Example prompt used for generating perspective-specific summaries. The System box details the task, guidelines, and input information, while the User box specifies the required output.