Enhancing Factuality and Transparency in Generative Models for Biomedical Question Answering

Ankita Behura, Siting Liang, Daniel Sonntag
German Research Center for Artificial Intelligence, Oldenburg University
Germany
{siting.liang, daniel.sonntag}@dfki.de
{siting.liang, daniel.sonntag}@uni-oldenburg.de

Abstract

Biomedical Question Answering (BQA) systems are vital for providing clinicians and researchers with efficient access to large amount of biomedical scientific studies. Existing automated BQA models, however, often rely on complex hybrid architectures to handle diverse question and answer formats, leading to inefficiency and high complexity. While domain-specific generative language models like BioBART offer a unified and simplified alternative capable of producing fluent human-like responses, they are prone to hallucination and lack interpretability, undermining their trustworthiness in critical healthcare domains. To address these limitations, this work introduces an enhanced model that augments BioBART with a pointer network for accurate token copying and a novel Keyphrase Filter (KPF) to guide attention toward critical information during generation. Experimental results on the BioASQ challenge demonstrate that the proposed Pointer-KPF model significantly outperforms the baseline BioBART, particularly on metrics for ideal answers. Furthermore, our evaluation shows that the model enhances transparency: pointer-guided attention heatmaps reveal improved input-output alignment, while keyphrase scores act as saliency maps to identify the most influential input segments. This approach not only reduces hallucination by strengthening textual grounding but also provides crucial insights into the model's reasoning, thereby increasing confidence and trust in its outputs.

Keywords: Biomedical Question Answering, Biomedical Language Model, Unified Question Answering

1. Introduction

Biomedical Natural Language Processing (BioNLP) plays a crucial role in improving information retrieval and processing within the biomedical domain by integrating expertise from bioinformatics, medicine, and artificial intelligence. Over the years, NLP techniques have been applied to classification, information extraction, question answering (QA), and drug discovery (Cohen and Hersh, 2005; Cohen and Demner-Fushman, 2014). Early approaches relied on semantic parsing technologies for identifying relations between disease (Mkrtchyan and Sonntag, 2014), rule-based techniques for negation handling, such as NegEx (Profitlich and Sonntag, 2021). Although ontology-based approaches (Sonntag et al., 2009, 2016) enhance biomedical entity recognition by incorporating structured knowledge, they suffer from high annotation costs and limited adaptability to evolving biomedical literature.

Traditional Biomedical Question Answering (BQA) systems have relied on information retrieval (IR) techniques (Voorhees, 2001; Niu et al., 2003), which, despite retrieving relevant text spans, struggle with complex medical queries, domain-specific terminology, and the need for precise, interpretable answers (Schmidt et al., 2016). To advance BQA research, the BioASQ challenge (Nentidis et al., 2023) provides expert-curated biomedical question answering datasets for model evaluation. As shown in Table 1, the questions cover a range of types, including **"yes/no"**, **"factoid"**, **"list"**, and **"summary"** questions. Answers are categorized as either **"exact"** (derived directly from the provided text) or **"ideal"** (human-like summaries of the relevant information).

Question Type	Example Question, Exact Answer, and Ideal Answer
Yes/No	**Question:** Proteomic analyses need prior knowledge of the organism's complete genome. Is the complete genome of the bacteria of the genus *Arthrobacter* available? **Exact Answer:** yes **Ideal Answer:** Yes, the complete genome sequence of *Arthrobacter* (two strains) is deposited in GenBank.
List	**Question:** List Hemolytic Uremic Syndrome Triad. **Exact Answer:** [anaemia, thrombocytopenia, renal failure] **Ideal Answer:** Hemolytic uremic syndrome (HUS) is a clinical syndrome characterized by the triad of anaemia, thrombocytopenia, renal failure.
Factoid	**Question:** What enzyme is inhibited by Opicapone? **Exact Answer:** [catechol-O-methyltransferase] **Ideal Answer:** Opicapone is a novel catechol-O-methyltransferase (COMT) inhibitor to be used as adjunctive therapy in levodopa-treated patients with Parkinson's disease.
Summary	**Question:** What kind of affinity purification would you use to isolate soluble lysosomal proteins? **Ideal Answer:** The rationale for purification of the soluble lysosomal proteins resides in their characteristic sugar, the mannose-6-phosphate (M6P), which allows an easy purification by affinity chromatography on immobilized M6P receptors.

Table 1: Examples of BioASQ Question Types with Exact and Ideal Answers for four question types. Exact answers (highlighted in red) are concise, fact-based responses directly extracted from input texts, typically named entities. Ideal answer generation in BQA aims to provide comprehensive, well-structured responses synthesizing relevant implicit knowledge.

Recent approaches leverage transformer-based models like BERT (Devlin et al., 2019) and its biomedical adaptations, such as BioBERT (Lee et al., 2020), and BioGPT (Luo et al., 2022), achieving strong performance on BQA tasks (Jin et al., 2022; Hu et al., 2023; Kim et al., 2023). However,

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 127–139
12 May 2026.

these models require separate fine-tuning for different question types, increasing training costs and system complexity. Additionally, scalability and interpretability remain significant challenges, limiting their clinical applicability (Jin et al., 2022; Lyu et al., 2024).

In this work, we propose a unified BQA system based on a domain-specific BioBART (Lewis et al., 2019; Yuan et al., 2022) model with multi-task learning to handle diverse question types within a Sequence to Sequence (Seq2Seq) architecture. Specifically, we integrate a pointer network (Vinyals et al., 2015; See et al., 2017) and a key phrase filtering mechanism into the Seq2Seq architecture. This improves the factual accuracy of generated answers by dynamically emphasizing relevant information during training and inference. Unlike existing key phrase labeling techniques (Liang et al., 2022), our approach enables end-to-end learning without requiring explicit key phrase annotations, making our system more efficient. Our experimental results demonstrate that incorporating a pointer network and key-phrase filtering improves both the precision of answers and the transparency of the BioBART model, thereby addressing critical challenges in BQA. Our goal is to foster trust and usability of pre-trained generative language models in clinical applications. We will publish our code and fine-tuned models.

2. Related Work

PLMs for Biomedical NLP. The scarcity of annotated biomedical data limits the performance of general-purpose language models in this specialized domain. Models like BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019), ClinicalBERT (Alsentzer et al., 2019), and PubMedBERT (Gu et al., 2021) have demonstrated superior performance in biomedical NLP tasks by effectively capturing domain-specific semantics. Models like BioELMo (Jin et al., 2019), and Bio-ELECTRA (Ozyurt, 2020) integrate domain-specific ontologies and knowledge graphs to enhance their understanding of biomedical concepts. Within the BioASQ challenge (Nentidis et al., 2022), PLMs-based extractive approaches have been widely adopted, with models such as BioM-ELECTRA (Alrowili and Vijay-Shanker, 2021), BioM-ALBERT (Alrowili, 2021), and Bio-ELECTRA (Ozyurt, 2020) achieving strong performance by treating BQA as a span prediction task (Jin et al., 2022). Fine-tuning on QA datasets has further enhanced these models, allowing them to learn task-specific answer extraction patterns (Nentidis et al., 2023).

Generative models for summary generation. Unlike extractive approaches that label text spans, generative models synthesize information from input sources to produce coherent, human-like responses. Encoder-decoder frameworks have become a standard approach for generating concise yet informative summaries across domains (Nallapati et al., 2017; Chopra et al., 2016). Transformer-based architectures such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) have been widely used in abstractive summarization tasks. BioBART (Yuan et al., 2022) adapts the BART architecture to the biomedical domain. BioGPT (Luo et al., 2022) is a decoder-only generative model designed for biomedical text generation. To ensure factual consistency with the source text, hybrid approaches combining extractive and abstractive techniques have been proposed (Gu et al., 2016; Kryściński et al., 2018). See et al. (2017) and Nallapati et al. (2016) integrated a pointer network (Vinyals et al., 2015) into an RNN-based encoder-decoder model, allowing the system to either generate words from a predefined vocabulary or copy words directly from the source text. Building on a BERT-based encoder-decoder model, Liang et al. (2022) integrated a pointer mechanism to enable direct copying of text spans from the source input, combined with span extraction supervision to enhance source text reconstruction during generation. Prompting large language models (Hsueh et al., 2023) has been explored to improve model performance. However, this approach requires key phrase annotations as generation targets during training. Despite these advancements, PLM-based methods face challenges in providing explainable answers, motivating further research into hybrid approaches for accountable biomedical QA.

3. Method

3.1. BioBART Baseline

We utilize BioBART (Yuan et al., 2022) as our baseline, a variant of BART (Lewis et al., 2020) pre-trained on biomedical corpora, including PubMed abstracts. This domain-specific pre-training enables BioBART to handle the complexities of biomedical vocabulary and concepts. BioBART employs a standard transformer encoder-decoder architecture with bidirectional encoder self-attention and causal decoder self-attention.

3.2. Pointer Network

To enhance BioBART's ability to leverage both generative and extractive capabilities for BQA, we integrated a pointer network as shown in Figure 1. The pointer network introduces an additional mechanism that adjusts the cross-attention scores so that the model can either copy words from the input

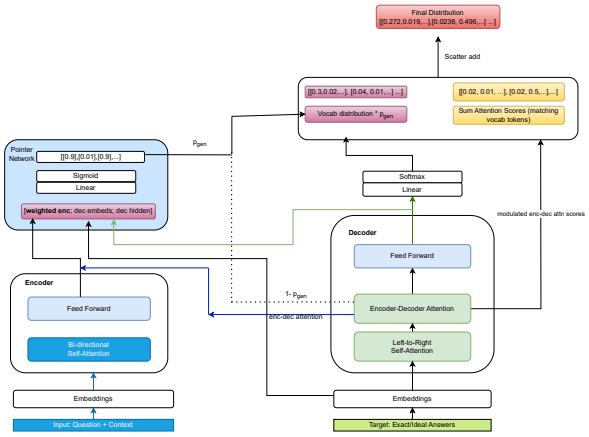

Figure 1: The pointer network runs in parallel to the standard BART generation process. It takes the decoder hidden state, the current input token embedding, and the weighted encoder hidden states as input. It produces a generation probability, p_{gen}, which is used directly to modulate the cross-attention scores. These modulated cross-attention scores are then summed to match the vocabulary tokens. The final output distribution is produced by combining the vocabulary distribution with these summed attention scores.

(close attention weight on source tokens) or generate new words (standard decoding). Specifically, the model dynamically decides between generating a token from the vocabulary or copying one from the input, using a learned probability distribution at each decoding step. At each decoding step t, the pointer network computes a generation probability p_{gen}, calculated as:

$$p_{\text{gen}} = \sigma(\mathbf{w}_{\text{ptr}}^T[h_t^x; y_t; s_t] + b_{\text{ptr}}) \tag{1}$$

where σ is the sigmoid function, $\mathbf{w}_{\text{ptr}}$ is a learnable weight vector, b_{ptr} is a bias term, $\mathbf{h}_{x_t}$ is the encoder hidden state, y_t is the current decoder input token embedding, and s_t is the decoder's hidden state. These hidden states from the final encoder are used as context vectors passed to each decoding step. Each decoder state s_t predicting the next token is also from the last decoder.

The attention-weighted encoder representation h_t^x is computed using the encoder-decoder cross-attention. Since the model uses multi-head attention, we compute the mean over all attention heads as suggested in Liang et al. (2022). Let $\alpha_{t,j}^{(i)}$ denote the attention weight from decoder step t to encoder token j for head i. The averaged attention is:

$$\bar{\alpha}_{t,j} = \frac{1}{N_{\text{heads}}} \sum_{i=1}^{N_{\text{heads}}} \alpha_{t,j}^{(i)} \tag{2}$$

The attended encoder representation is then:

$$h_t^x = \sum_{j=1}^{T_x} \bar{\alpha}_{t,j} \cdot h_j^x \tag{3}$$

where j indexes the source tokens and T_x is the input sequence length.

The pointer network modulates the cross-attention scores by scaling them with $(1 - p_{\text{gen}})$. The final output token distribution $P_{\text{final}}(w)$ is then computed as:

$$P_{\text{final}}(w) = p_{\text{gen}} \cdot P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \cdot \sum_{j:x_j=w} \bar{\alpha}_{t,j} \tag{4}$$

where $P_{\text{vocab}}(w)$ is the vocabulary distribution, and the second term aggregates attention scores over all source positions j where the input token x_j matches the vocabulary token w. The benefit of incorporating a pointer network is to improve the alignment between the generated output and the input context, reducing factual inconsistencies.

3.3. Keyphrase Filter

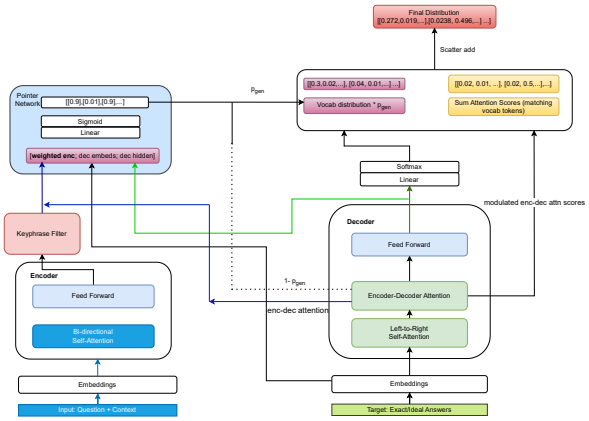

Figure 2: The KPF module is implemented as a linear layer that predicts keyphrase scores for each input token using the encoder hidden states. These scores are used to weight the encoder hidden states, which are then employed in the cross-attention mechanism and as input to the pointer network. The cross-attention weights are also weighted further by the keyphrase predictions.

To ensure that the model focuses on crucial information in the input text, we introduce a Keyphrase Filter (KPF) between the encoder and decoder. See Figure 2 for an architectural overview of the Pointer-KPF network. The KPF is implemented as a linear layer (binary classifier) trained jointly with the model. It predicts keyphrase scores based on encoder hidden states and is trained implicitly to identify important keyphrases without explicit annotations.

For each encoder token position j, the KPF predicts a score k_j using:

$$e_j = \text{KPF}(h_j^x) \tag{5}$$

$$k_j = \sigma(e_j) \tag{6}$$

where σ is the sigmoid function.

The keyphrase scores are used to modulate the cross-attention weights before aggregation. Specifically, the attention weights are re-weighted as:

$$\tilde{\alpha}_{t,j} = k_j \cdot \bar{\alpha}_{t,j} \tag{7}$$

The modified attended encoder representation is then:

$$h_t^{x'} = \sum_{j=1}^{T_x} \tilde{\alpha}_{t,j} \cdot h_j^x \tag{8}$$

These modified attention scores are also incorporated into the pointer mechanism. The final output distribution becomes:

$$P_{\text{final}}(w) = p_{\text{gen}} \cdot P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \cdot \sum_{j:x_j=w} k_j \cdot \bar{\alpha}_{t,j} \tag{9}$$

The KPF assigns the saliency of each token in the input sequence, which is then used to modulate both the encoder hidden states and the cross-attention scores in the decoder. This mechanism guides the model to prioritize relevant keyphrases while reducing the influence of less informative tokens, ultimately strengthening its extractive capabilities and improving the accuracy of generated answers. While the pointer network ensures that factual accuracy is preserved by copying key information from the input, the KPF reinforces this process by explicitly guiding attention toward keyphrases in the input text.

4. Experiments

We evaluated our models using the 11th release of BioASQ dataset (Nentidis et al., 2023). The BioASQ dataset provides expert-curated training questions, relevant abstracts, snippets, and correct answers. Table 2 presents the final distribution of question types across training, validation, and test sets.

4.1. Data Pre-processing

We adopted the "Snippet as is" pre-processing strategy, which utilized individual-provided snippets as contexts for each question, without further concatenation or processing. This approach was selected due to its balanced performance across all question types (Yes/No, List, Factoid, and Summary), as demonstrated by Yoon et al. (2019). This strategy offers data augmentation, focused attention on crucial terms, enhanced semantic nuance understanding, and promotes answer diversity. The preprocessing extracts a single snippet per question, preserving its original structure across the dataset.

Split		Yes/No	List	Factoid	Summary
Train		25,010	13,817	26,685	12,997
Val		20,007	10,878	17,790	8,665
	Set 1	319	432	591	326
	Set 2	269	210	660	655
Test	Set 3	378	310	400	341
	Set 4	327	150	414	318
	Set 5	596	499	527	293

Table 2: Data Statistics for Train, Validation, and Test Sets based on the four question types.

4.2. Task-Specific Token

Adapting pretrained language models (PLMs) to downstream tasks often involves addressing variations in data structure and desired output formats. While post-processing techniques are common, they can be tedious and may not generalize well. To address the diverse question types and answer formats, we implemented a task-specific token strategy. Inspired by prior work (Ateia and Kruschwitz, 2023; Achiam et al., 2023), we prepend task-specific tokens to the input sequence, explicitly informing the model about the question type and expected answer format, see Table 3. This enables the model to learn distinct answer patterns associated with specific question types during training.

Answer Format	Question Type	Task-Specific Token
Exact	Factoid	`<factoid_exact_answer>`
	List	`<list_exact_answer>`
	Yes/No	`<yesno_exact_answer>`
Ideal	Factoid	`<factoid_ideal_answer>`
	List	`<list_ideal_answer>`
	Yes/No	`<yesno_ideal_answer>`
	Summary	`<summary_ideal_answer>`

Table 3: Task-specific tokens used for different question types and answer formats.

4.3. Training and Inference

We evaluated BioBART, BioBART with a pointer network (Pointer-Only), BioBART with a pointer network and keyphrase filter (Pointer-KPF), and removing the pointer network from the Pointer-KPF model

(KPF-Only). All models were initialized with the pre-trained BioBART[1] model (Yuan et al., 2022). We train the models on a single NVIDIA A100-SXM4-80GB GPU, with a batch size of 16, a maximum input length of 512, a maximum of 3 training epochs, and with a learning rate of 2×10^{-5}.

4.4. Evaluation Metrics

To evaluate the robustness and consistency of our models, we examined their performance across the five test sets. For each model, question type, and test set, we generated multiple answers while systematically varying the beam size from 1 to 5. The BioASQ challenge evaluates exact answers differently based on the question type. Set N as the total number of questions.

Strict Accuracy (SAcc) SAcc is the metric used for factoid questions. It measures the proportion of questions for which the correct answer is predicted and placed at the top of the ranked list of answers:

$$SAcc = \frac{c_1}{N} \tag{10}$$

where c_1 is the number of questions answered correctly when only the top answer is considered.

Lenient Accuracy (LAcc) LAcc is a metric that increases when the correct answer is predicted and placed anywhere within the top k answers:

$$LAcc = \frac{c_k}{N} \tag{11}$$

where c_k is the number of questions answered correctly when all top k answers are considered.

Mean Reciprocal Rank (MRR) MRR is a metric that evaluates the effectiveness of a system in ranking correct answers. It increases inversely with the position of the correct answer in the ranked list:

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i} \tag{12}$$

where $rank_i$ is the position of the correct answer for the i-th question. In the BioASQ challenge, the top 5 (k = 5) answers are predicted for each question, and MRR is calculated based on these predictions.

Precision, Recall, and F1-Measure For list questions, the evaluation is based on precision (P), recall (R), and F1-measure ($F1$). These metrics are averaged across all questions to compute the Mean Average Precision (MAP), Mean Average Recall (MAR), and Mean Average F1-measure (MAF).

[1] GanjinZero/biobart-v2-base

The formulas for precision, recall, and F1-measure are:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{15}$$

where TP is the number of correct answers predicted, FP is the number of incorrect answers predicted, FN is the number of correct answers not predicted.

ROUGE Metrics ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) measures the overlap between system-generated and reference (golden) ideal answers. The metric applied in our evaluation is:

- ROUGE-1: Evaluates unigram overlap between the generated and reference answers.

5. Results

5.1. Results of Exact Answer Generation

Table 4 presents the exact answer generation performance of each model across the test sets. The mean reflects the average performance across the generations with varying beam sizes, while the standard deviation indicates the variability in performance across these generations. A smaller standard deviation signifies more stable performance across different beam sizes.

Unlike Factoid and List questions, all models perform well in Yes/No questions, with the Baseline model already achieving accuracy (>69%). Pointer-KPF and Pointer-Only demonstrate the benefits of pointer networks, but KPF-Only surprisingly performs best in some test batches, in particular for Factoid and Yes/No questions, suggesting a complex interaction between pointer networks and keyphrase filtering. Pointer-Only consistently outperforms other models in MAP and MAF for List questions, demonstrating the effectiveness of Pointer Networks for list-type answers. Pointer-KPF exhibits a substantial performance boost, achieving up to a 50x increase in MAP (test set 2, from 0.005 to 0.202). Keyphrase filtering also contributes, but its impact is less consistent.

5.2. Results of Ideal Answer Generation

For each question type (yes/no, factoid, list, summary), systems are expected to provide an "ideal" answer resembling a concise response that a

Test Set	Model	Factoid			List			Yes/No	
		LAcc	MRR	SAcc	MAP	MAR	MAF	Acc	MAF
1	Baseline	0.200 ± 0.013	0.160 ± 0.027	0.135 ± 0.033	0.077 ± 0.011	0.089 ± 0.013	0.082 ± 0.012	0.696 ± 0.000	0.582 ± 0.000
	Pointer-Only	0.294 ± 0.021	0.264 ± 0.023	0.235 ± 0.021	0.410 ± 0.006	0.480 ± 0.009	0.411 ± 0.007	$\mathbf{0.774 \pm 0.020}$	$\mathbf{0.633 \pm 0.049}$
	Pointer-KPF	$\mathbf{0.318 \pm 0.025}$	$\mathbf{0.296 \pm 0.016}$	$\mathbf{0.282 \pm 0.016}$	$\mathbf{0.445 \pm 0.019}$	$\mathbf{0.479 \pm 0.023}$	$\mathbf{0.446 \pm 0.018}$	0.739 ± 0.000	0.425 ± 0.000
	KPF-Only	0.276 ± 0.039	0.257 ± 0.040	0.247 ± 0.040	0.357 ± 0.033	0.442 ± 0.052	0.366 ± 0.035	0.739 ± 0.000	0.546 ± 0.000
2	Baseline	0.288 ± 0.025	0.263 ± 0.026	0.241 ± 0.032	0.005 ± 0.007	0.007 ± 0.007	0.006 ± 0.007	0.778 ± 0.000	0.723 ± 0.000
	Pointer-Only	0.459 ± 0.034	0.382 ± 0.026	0.335 ± 0.033	$\mathbf{0.260 \pm 0.018}$	$\mathbf{0.191 \pm 0.007}$	$\mathbf{0.207 \pm 0.012}$	0.767 ± 0.025	0.669 ± 0.022
	Pointer-KPF	0.453 ± 0.034	0.379 ± 0.035	0.341 ± 0.033	0.202 ± 0.023	0.144 ± 0.010	0.157 ± 0.014	0.744 ± 0.031	0.606 ± 0.067
	KPF-Only	$\mathbf{0.465 \pm 0.025}$	$\mathbf{0.394 \pm 0.023}$	$\mathbf{0.353 \pm 0.029}$	0.205 ± 0.006	0.174 ± 0.004	0.179 ± 0.004	$\mathbf{0.833 \pm 0.000}$	$\mathbf{0.778 \pm 0.000}$
3	Baseline	0.238 ± 0.035	0.176 ± 0.035	0.150 ± 0.035	0.061 ± 0.003	0.058 ± 0.002	0.059 ± 0.002	0.792 ± 0.000	0.705 ± 0.000
	Pointer-Only	$\mathbf{0.325 \pm 0.036}$	$\mathbf{0.268 \pm 0.042}$	$\mathbf{0.244 \pm 0.041}$	$\mathbf{0.225 \pm 0.007}$	0.213 ± 0.006	$\mathbf{0.214 \pm 0.006}$	$\mathbf{0.816 \pm 0.037}$	$\mathbf{0.747 \pm 0.070}$
	Pointer-KPF	0.275 ± 0.074	0.214 ± 0.059	0.194 ± 0.051	0.138 ± 0.008	0.129 ± 0.009	0.130 ± 0.008	0.725 ± 0.023	0.561 ± 0.055
	KPF-Only	0.287 ± 0.034	0.237 ± 0.034	0.219 ± 0.038	0.218 ± 0.026	$\mathbf{0.221 \pm 0.010}$	0.214 ± 0.018	0.792 ± 0.000	0.705 ± 0.000
4	Baseline	0.310 ± 0.029	0.274 ± 0.027	0.252 ± 0.027	0.091 ± 0.020	0.083 ± 0.014	0.083 ± 0.016	0.792 ± 0.000	0.736 ± 0.000
	Pointer-Only	0.374 ± 0.018	0.335 ± 0.017	0.310 ± 0.018	0.247 ± 0.010	0.210 ± 0.016	0.219 ± 0.012	0.875 ± 0.000	0.845 ± 0.006
	Pointer-KPF	0.361 ± 0.027	0.303 ± 0.037	0.265 ± 0.058	$\mathbf{0.290 \pm 0.011}$	$\mathbf{0.252 \pm 0.007}$	$\mathbf{0.264 \pm 0.008}$	0.900 ± 0.023	0.870 ± 0.026
	KPF-Only	$\mathbf{0.413 \pm 0.014}$	$\mathbf{0.381 \pm 0.015}$	$\mathbf{0.361 \pm 0.027}$	0.264 ± 0.013	0.226 ± 0.004	0.237 ± 0.007	$\mathbf{0.917 \pm 0.000}$	$\mathbf{0.889 \pm 0.000}$
5	Baseline	0.172 ± 0.024	0.133 ± 0.028	0.103 ± 0.035	0.138 ± 0.007	0.137 ± 0.006	0.137 ± 0.006	0.714 ± 0.000	0.689 ± 0.000
	Pointer-Only	$\mathbf{0.214 \pm 0.015}$	0.165 ± 0.012	0.131 ± 0.016	$\mathbf{0.380 \pm 0.004}$	$\mathbf{0.386 \pm 0.003}$	$\mathbf{0.378 \pm 0.003}$	0.786 ± 0.000	0.775 ± 0.000
	Pointer-KPF	0.200 ± 0.016	0.165 ± 0.005	$\mathbf{0.138 \pm 0.000}$	0.342 ± 0.020	0.337 ± 0.018	0.337 ± 0.020	0.743 ± 0.039	0.714 ± 0.056
	KPF-Only	0.207 ± 0.000	$\mathbf{0.167 \pm 0.000}$	0.138 ± 0.000	0.341 ± 0.017	0.347 ± 0.019	0.340 ± 0.017	$\mathbf{0.786 \pm 0.000}$	$\mathbf{0.775 \pm 0.000}$

Table 4: Performance metrics for different models for exact answer generation. Values are reported as Mean ± Std across varying beam sizes.

Test Set	Model	Factoid	List	Yes/No	Summary
1	Baseline	0.361 ± 0.008	0.163 ± 0.023	0.457 ± 0.039	0.414 ± 0.020
	Pointer-only	$\mathbf{0.383 \pm 0.031}$	0.360 ± 0.017	0.476 ± 0.027	0.473 ± 0.036
	Pointer-KPF	0.378 ± 0.029	$\mathbf{0.441 \pm 0.056}$	$\mathbf{0.477 \pm 0.011}$	$\mathbf{0.509 \pm 0.032}$
	KPF-Only	0.331 ± 0.023	0.401 ± 0.067	0.459 ± 0.070	0.429 ± 0.043
2	Baseline	0.362 ± 0.018	0.098 ± 0.020	0.445 ± 0.029	0.416 ± 0.011
	Pointer-only	0.390 ± 0.011	0.289 ± 0.031	0.445 ± 0.040	0.447 ± 0.043
	Pointer-KPF	$\mathbf{0.409 \pm 0.004}$	$\mathbf{0.313 \pm 0.022}$	$\mathbf{0.455 \pm 0.017}$	$\mathbf{0.473 \pm 0.061}$
	KPF-Only	0.378 ± 0.028	0.3 ± 0.027	0.428 ± 0.057	0.392 ± 0.050
3	Baseline	0.449 ± 0.028	0.113 ± 0.009	0.393 ± 0.008	0.400 ± 0.017
	Pointer-only	0.457 ± 0.009	0.273 ± 0.007	0.405 ± 0.014	0.435 ± 0.003
	Pointer-KPF	$\mathbf{0.467 \pm 0.016}$	$\mathbf{0.299 \pm 0.043}$	0.406 ± 0.013	$\mathbf{0.457 \pm 0.020}$
	KPF-Only	0.444 ± 0.053	0.283 ± 0.037	$\mathbf{0.455 \pm 0.060}$	0.443 ± 0.014
4	Baseline	0.361 ± 0.028	0.077 ± 0.006	0.356 ± 0.017	0.359 ± 0.011
	Pointer-only	0.377 ± 0.010	0.271 ± 0.008	0.395 ± 0.043	$\mathbf{0.462 \pm 0.034}$
	Pointer-KPF	$\mathbf{0.412 \pm 0.013}$	$\mathbf{0.337 \pm 0.064}$	0.391 ± 0.028	0.452 ± 0.032
	KPF-Only	0.388 ± 0.041	0.264 ± 0.014	$\mathbf{0.416 \pm 0.058}$	0.384 ± 0.009
5	Baseline	0.403 ± 0.008	0.145 ± 0.010	0.354 ± 0.043	0.367 ± 0.020
	Pointer-only	0.393 ± 0.011	0.353 ± 0.003	0.381 ± 0.010	0.377 ± 0.010
	Pointer-KPF	$\mathbf{0.432 \pm 0.052}$	0.350 ± 0.009	$\mathbf{0.430 \pm 0.008}$	$\mathbf{0.429 \pm 0.020}$
	KPF-Only	0.376 ± 0.034	$\mathbf{0.357 \pm 0.024}$	0.418 ± 0.059	0.354 ± 0.021

Table 5: ROUGE-1 F1-scores (Mean ± Std) of varying beam sizes. Bold values indicate the best mean score within each batch.

biomedical expert might provide. Table 5 presents ROUGE-1 F1-scores for ideal answer generation across different question types (Factoid, List, Yes/No, and Summary) and all test sets.

For factoid questions, performance is generally similar across models, though the Pointer-KPF model tends to perform best. This suggests that the combination of the pointer network and keyphrase filtering yields the best results for this question type. Pointer-Only consistently shows improvement over both Baseline and KPF-Only models, indicating that the pointer network plays a crucial role in enhancing factoid answer generation.

For list questions, Pointer-KPF frequently achieved the best ROUGE-1 scores, leveraging both keyphrase filtering and the pointer network. Notably, it showed a substantial improvement over the Baseline.

For yes/no questions, performance differences were minimal, suggesting BioBART's inherent effectiveness for these simpler questions. For factoid questions, Pointer-KPF generally achieved the highest scores, demonstrating the complementary benefits of pointer networks and keyphrase filtering. Pointer-Only consistently outperformed the Baseline, highlighting the importance of the pointer network. However, in Test Set 1, the Baseline slightly surpassed Pointer-Only, underscoring dataset-specific performance variations.

For summary questions, Pointer-KPF consistently demonstrated the strongest performance, reinforcing trends observed in other question types. Pointer-Only also outperformed the Baseline, further highlighting the pointer network's role in improving long-form answer generation.

Our analysis of ideal answer generation revealed several key findings:

- Synergistic Effect of Pointer Network and Keyphrase Filtering: Pointer-KPF, which combines both mechanisms, generally achieved the strongest performance across factoid, list, and summary questions, demonstrating the benefits of this combined approach.

- Effectiveness in List Questions: Significant performance improvements were observed for list questions, particularly with Pointer-KPF, highlighting the effectiveness of both pointer networks and keyphrase filtering for this question type.

- While Pointer-KPF often achieves the highest scores, the performance across the three model settings is relatively close. In some test sets, KPF-Only even surpasses Pointer-Only, albeit by a small margin.

5.3. Impact of beam size on Answer Generation

To further investigate the impact of decoding parameters on Pointer-KPF model performance, we conducted beam size experiments, varying the beam size from 1 to 5. This allowed us to explore the sensitivity of our models to the breadth of the search space during answer generation for both exact and ideal answers. The results show that the impact of beam size on exact answer generation is minimal (Figure 3). Performance remains relatively stable across different beam sizes, with only slight fluctuations in metrics like factoid lenient accuracy and list MAP/F1. This consistent trend suggests that the model's ability to identify exact answer spans is not significantly affected by increasing the beam size. The model appears to identify relevant answer spans with relatively small beam sizes. One possible explanation is that the search space for exact matches is inherently smaller and more focused than that for ideal matches, making the beam size less influential.

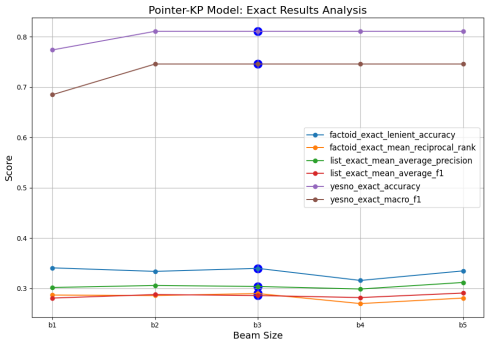

Figure 3: Pointer-KPF - Exact Answers

For ideal answer generation, the impact of beam size is more pronounced (Figure 4). For instance, the Pointer-KPF model's list ROUGE-1 improves from 0.297 at beam size 1 to 0.340 at beam size 3. This suggests that exploring a wider range of paraphrased answers and keyphrase combinations with larger beam sizes is particularly beneficial for generating better ideal answers, especially for list questions. This is likely because the search space for ideal matches is significantly larger and more diverse, requiring a broader search to identify optimal combinations of paraphrased text and keyphrases.

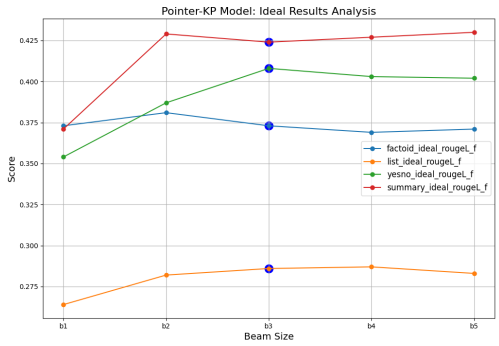

Figure 4: Pointer-KPF - Ideal Answers

6. Interpretability

Question	Which drugs are included in the CABENUVA pill?
Context	A regimen comprising extended release injectable suspensions of cabotegravir and rilpivirine for concurrent administration (CABENUVA™) is being developed by ViiV Healthcare and Janssen Pharmaceutica (Janssen) as a complete regimen for HIV infection.
Ideal Answer Reference	CABENUVA contains cabotegravir and rilpivirine. It is used for treatment of HIV.
Baseline	CABENUVA pill includes travaginal progesterone. It is approved for use in HIV-infected patients in the USA and Canada.
Pointer	CABNERUVA includes cabotegravir, rilpivirine, and tmburt, for the concurrent treatment of HIV infection.
Pointer-KPF	CABENUVA includes cabotegravir and rilpivirine for concurrent administration to HIV-infected patients.

Table 6: Comparison of generated ideal answers of different model settings for a list-type question. Text highlighted in red indicates a factual error or hallucination.

To understand performance improvements, we analyzed the model generations and attention patterns for a list-type question. Firstly, Table 6 presents the generations of different model settings and the reference, illustrating the progressive improvement from the Baseline BioBART model to the Pointer-KPF model. The Baseline model fails, hallucinating an unrelated drug ("travaginal progesterone"). The Pointer model shows improved factual grounding, correctly identifying the two primary drugs, but introduces new errors, including a misspelling of the product name ("CABNERUVA") and an invented component ("tmburt"). The Pointer-

KPF model generates the most accurate and faithful response, correctly stating the drug components.

Baseline model generated an incorrect answer, relying on generic vocabulary, as evidenced by its diffused cross-attention shown in Figure 6 (in the Appendix). The Pointer model improved, identifying correct components, but hallucinated an irrelevant term, displaying broader attention (see Figure 7). Conversely, the Pointer-KPF model generated accurate answers, demonstrating the KPF's ability to focus on relevant keyphrases, confirmed by their concentrated attention (Figure 8).

However, while cross-attention heatmaps provide useful insights, they have limitations: they may not fully capture model reasoning due to averaging across layers and attention heads, they carry a risk of misinterpreting high attention as meaningful contribution, and they are not intuitive for non-ML users. To address these issues and improve interpretability, particularly regarding keyphrases, we utilize saliency maps, specifically keyphrase importance scores derived from the KPF module (Figure 5). These scores highlight input tokens receiving the most attention, providing a more human-readable interpretation than cross-attention alone. The KPF scores, as seen in Figure 5, demonstrate the model's focus on relevant input parts, with higher saliency indicating more important keyphrases. Compared to heatmaps, these scores offer a clearer and more intuitive representation of model behavior, enabling more direct tracing of how specific keyphrases contribute to the final prediction. This visualization offers a more accessible way to understand the model's decision-making process, especially for non-expert users.

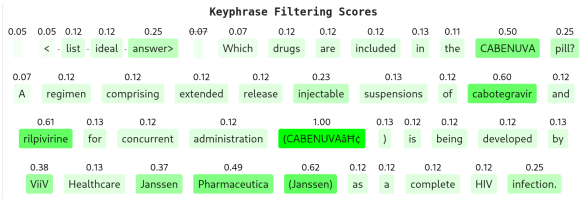

Figure 5: Keyphrase saliency scores overlaid on the input sequence for the list-type question. Color gradient indicates keyphrase saliency.

6.1. Comparison with BioASQ Leaderboard Systems

The BioASQ challenge remains the definitive benchmark for evaluating semantic question-answering capabilities in the biomedical domain. By examining the performance of the Pointer-KPF model relative to the official results from BioASQ 11b (2023), 12b (2024), and the most recent 13b (2025–2026) batches, the improvements of the proposed architectural enhancements can be contextualized within the broader state-of-the-art. Table 7 compares our self-interpretable Pointer-KPF model with recent large language model (LLM)-based approaches across multiple BioASQ editions. LLM systems such as IISR-1 based on (Hsueh et al., 2023) and UR-IW-1 using Mistral-7B, as well as fine-tuned and prompt-based variants (Kim et al., 2025), consistently achieve near-perfect Yes/No accuracy and strong performance on factoid and list questions, benefiting from their large parameter scales (7B–1.8T). Similarly, domain-specific models like BioLinkBERT (Kim et al., 2023) demonstrate competitive results with fewer parameters (340M), highlighting the advantage of biomedical pretraining. Frontier LLMs and heavily ensembled systems demonstrate superior absolute scores on the BioASQ 11b and 12b leaderboards. Unlike these LLM-based systems, which function largely as black boxes, Pointer-KPF, extending the previous research work, introduces an explicit, self-interpretable keyphrase filtering mechanism. This enables transparent reasoning by highlighting task-relevant keyphrases, offering a clear trade-off between performance and interpretability. For instance, our model's averaged Factoid MRR of 0.303 is comparable to results seen in few-shot prompting of much larger models, such as the Mistral-7B 10-shot system (MRR 0.346) in the 13b challenge. Crucially, the Pointer-KPF achieves this performance with a BioBART-base backbone (140M parameters), which is significantly more efficient than the 7B–70B parameter models currently dominating the leaderboard. The results highlight an important direction for future research: bridging the gap between interpretability and performance.

System / Model Category	Edition	Yes/No Acc.	Factoid MRR	List F1	Params
IISR-1 (GPT-4)	11b (2023)	1.0000	0.4211	0.7043	∼1.8T
DMIS-KU-5 (BioLinkBERT)	11b (2023)	0.9583	0.6053	0.7219	340M
UR-IW-1 (Mistral-7B)	12b (2024)	1.0000	0.3254	0.4266	7.3B
Mistral-7B (Fine-tuned)	12b (2024)	0.9600	0.2381	0.5265	7.3B
2025-DMIS-KU-2	13b (2025)	1.0000	0.5321	0.5322	∼7B
Mistral-7B-Ins (10-shot)	13b (2025)	1.0000	0.3462	0.2374	7.3B
Pointer-KPF (this study)	11b (2023)	0.779*	0.303*	0.254*	140M

Table 7: Performance comparison across BioASQ editions. *Pointer-KPF results are averaged over five BioASQ 11b test sets.

7. Conclusion

In this work, we aim to demonstrate that a single, generative model could effectively handle diverse question types and answer formats, overcoming the limitations of complex, multi-model architectures. To achieve this, we introduce two key enhance-

ments: a pointer network that selectively copies tokens from the input context, modulating cross-attention scores to bridge abstractive summarization and precise information extraction, and a KPF module that predicts keyphrase scores to guide attention towards salient information, refining the model's ability to identify and utilize crucial context.

For exact answers, we observe substantial performance gains, particularly in list-type questions, with a consistent 20% increase in Mean Average F1 (MAF) scores using the pointer network. Factoid questions also highlight the importance of pointer mechanisms, with Pointer-KPF and Pointer-Only models demonstrating superior extractive capabilities. In ideal answer generation, the Pointer-KPF model consistently outperforms the baseline model, showcasing enhanced information extraction and paraphrasing abilities. Furthermore, to enhance interpretability and user confidence, we apply keyphrase saliency maps, providing user-friendly visualizations of the model's reasoning. In summary, our work demonstrates the effectiveness of integrating pointer networks and KPF into BioBART for enhanced BQA, highlighting the potential of refining generative models in biomedical information extraction.

While Pointer-KPF operates with a significantly smaller parameter size (140M) and does not yet match the raw performance of LLM-based systems, it establishes a strong foundation for interpretable QA. Future work will explore further refinements of the KPF module and evaluate performance on a broader range of biomedical tasks and datasets. Further improvements are needed to enhance answer accuracy while preserving the model's ability to provide clear, faithful explanations, potentially through better integration with pretrained knowledge, more robust keyphrase extraction, or hybrid approaches that combine interpretability with the strengths of large-scale models.

Limitations

While our research demonstrated promising results, several limitations warrant further investigation. The potential inadequacy of ROUGE scores in capturing semantic correctness and the sometimes misleading nature of attention visualizations suggest a need for alternative evaluation and analysis methods. To address these limitations, future work should focus on developing interactive keyphrase interfaces to enhance user trust, conducting user studies to assess keyphrase interactivity, implementing interactive mechanisms for keyphrase manipulation, exploring advanced attention analysis and decoding strategies, and leveraging extended context utilization with advanced data augmentation techniques. Ultimately, these efforts aim to enhance model performance, improve interpretability, and foster a more robust and reliable biomedical question answering system.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant numbers 01IW24006 (NoIDLEChatGPT), as well as by the Endowed Chair of Applied AI at the University of Oldenburg.

Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sultan Alrowili. 2021. Large biomedical question answering models with albert and electra.

Sultan Alrowili and K Vijay-Shanker. 2021. Biomtransformers: building large biomedical language models with bert, albert and electra. In *Proceedings of the 20th workshop on biomedical language processing*, pages 221–227.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Samy Ateia and Udo Kruschwitz. 2023. Is chatgpt a biomedical expert. *Exploring the Zero-Shot Performance of Current GPT Models in Biomedical Tasks*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Aaron M Cohen and William R Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71.

Kevin Bretonnel Cohen and Dina Demner-Fushman. 2014. *Biomedical natural language processing*, volume 11. John Benjamins Publishing Company.

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.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Chun-Yu Hsueh, Yu Zhang, Yu-Wei Lu, Jen-Chieh Han, Wilailack Meesawad, and Richard Tzong-Han Tsai. 2023. Ncu-iisr: Prompt engineering on gpt-4 to stove biological problems in bioasq 11b phase b. In *11th BioASQ Workshop at the 14th Conference and Labs of the Evaluation Forum (CLEF)*.

Zhongjian Hu, Peng Yang, Bing Li, Yuankang Sun, and Biao Yang. 2023. Biomedical extractive question answering based on dynamic routing and answer voting. *Information Processing & Management*, 60(4):103367.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 82–89.

Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Huaiyuan Ying, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. 2022. Biomedical question answering: a survey of approaches and challenges. *ACM Computing Surveys (CSUR)*, 55(2):1–36.

Hajung Kim, Hoonick Lee, Yewon Cho, Jungwoo Park, Jueon Park, Soyon Park, Yan Ting Chok, Seungheun Baek, Donghyeon Lee, and Jaewoo Kang. 2025. Prompting matters: snippet-aware strategies for biomedical qa with llms in bioasq 13b. In *CLEF*.

Hyunjae Kim, Hyeon Hwang, Chaeeun Lee, Minju Seo, Wonjin Yoon, and Jaewoo Kang. 2023. Exploring approaches to answer biomedical questions: From pre-processing to gpt-4.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Siting Liang, Klaus Kades, Matthias Fink, Peter Full, Tim Weber, Jens Kleesiek, Michael Strube, and Klaus Maier-Hein. 2022. Fine-tuning bert models for summarizing german radiology findings. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 30–40.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

Daoming Lyu, Xingbo Wang, Yong Chen, and Fei Wang. 2024. Language model and its interpretability in biomedicine: A scoping review. *Iscience*.

Tigran Mkrtchyan and Daniel Sonntag. 2014. Deep parsing at the clef2014 ie task (dfki-medical). In *CEUR Workshop Proceedings*, volume 1180, pages 138–146.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016. Classify or select: Neural architectures for extractive document summarization. *CoRR*, abs/1611.04244.

Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, Salvador Lima López, Eulália Farré-Maduell, Luis Gasco, Martin Krallinger, and Georgios Paliouras. 2023. Overview of bioasq 2023: The eleventh bioasq challenge on large-scale biomedical semantic indexing and question answering. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 227–250. Springer.

Anastasios Nentidis, Georgios Katsimpras, Eirini Vandorou, Anastasia Krithara, Antonio Miranda-Escalada, Luis Gasco, Martin Krallinger, and Georgios Paliouras. 2022. Overview of bioasq 2022: The tenth bioasq challenge on large-scale biomedical semantic indexing and question answering. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 337–361. Springer.

Yun Niu, Graeme Hirst, Gregory McArthur, and Patricia Rodriguez-Gianolli. 2003. Answering clinical questions with role identification. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, pages 73–80.

Ibrahim Burak Ozyurt. 2020. On the effectiveness of small, discriminatively pre-trained language representation models for biomedical text mining. *bioRxiv*, pages 2020–05.

Hans-Jürgen Profitlich and Daniel Sonntag. 2021. A case study on pros and cons of regular expression detection and dependency parsing for negation extraction from german medical documents. technical report. *arXiv preprint arXiv:2105.09702*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Danilo Schmidt, Hans-Jürgen Profitlich, and Daniel Sonntag. 2016. Towards integrated information extraction and facetted search applications in nephrology. In *EMSA-RMed@ ESWC*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Daniel Sonntag, Volker Tresp, Sonja Zillner, Alexander Cavallaro, Matthias Hammon, André Reis, Peter A Fasching, Martin Sedlmayr, Thomas Ganslandt, Hans-Ulrich Prokosch, et al. 2016. The clinical data intelligence project: a smart data initiative. *Informatik-Spektrum*, 39:290–300.

Daniel Sonntag, Pinar Wennerberg, Paul Buitelaar, and Sonja Zillner. 2009. Pillars of ontology treatment in the medical domain. *Journal of Cases on Information Technology (JCIT)*, 11(4):47–73.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Ellen M Voorhees. 2001. The trec question answering track. *Natural Language Engineering*, 7(4):361–378.

Wonjin Yoon, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong, and Jaewoo Kang. 2019. Pre-trained language model for biomedical question answering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 727–740. Springer.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022. Biobart: Pretraining and evaluation of a biomedical generative language model. *arXiv preprint arXiv:2204.03905*.

Appendix

In the Appendix, Figures 6, 7, and 8 provide a comparative visualization of cross-attention patterns across the three models. These attention weights represent the strength of interaction between a specific input and output token. It is indicated by the intensity of the cell in which they meet. Higher intensity values correspond to stronger attention, revealing which input elements influence the model's predictions the most at each decoding step. By examining these patterns, we can gain insight into the model's reasoning process, identify whether it focuses on relevant information.

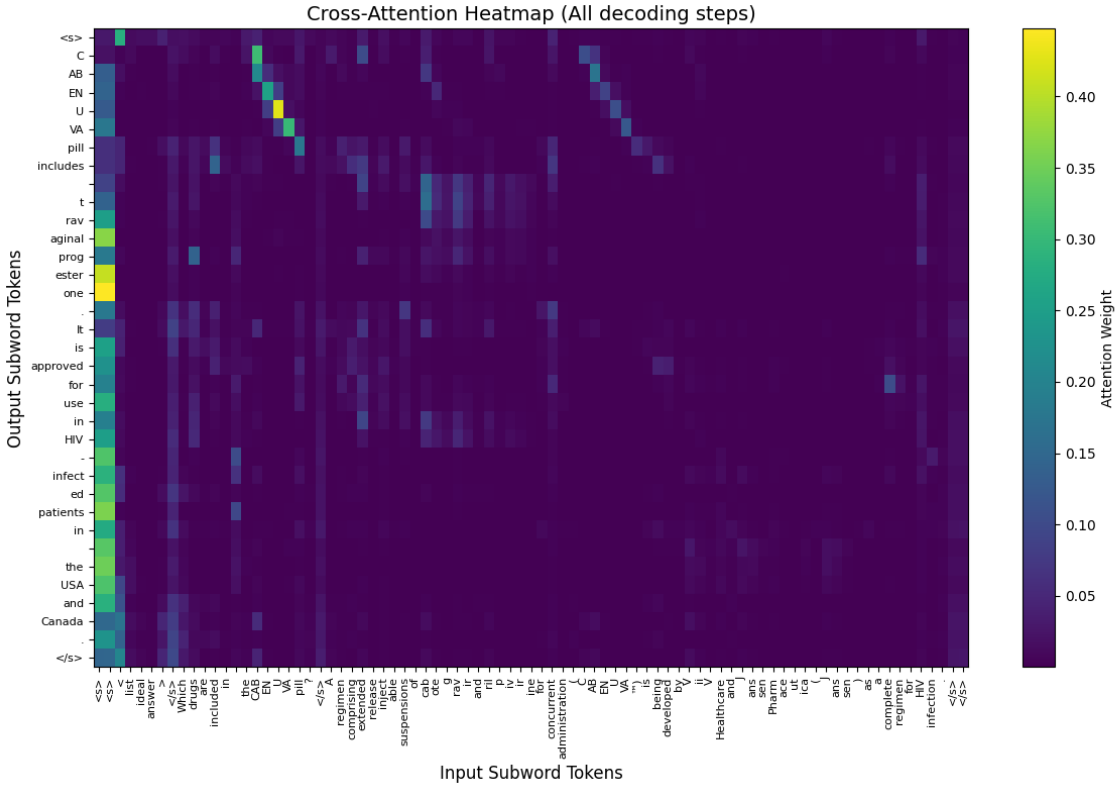

Figure 6: Baseline Cross-Attention Heatmap.

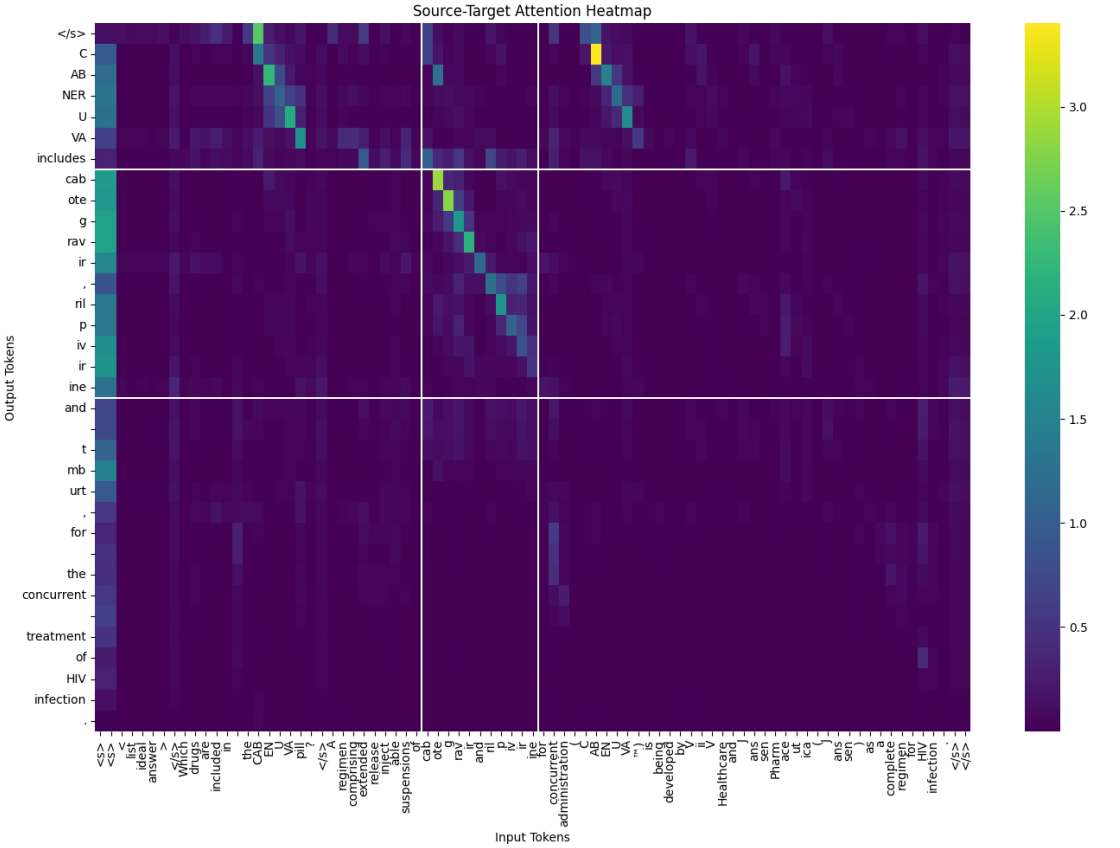

Figure 7: Cross-attention heatmaps for Pointer-only generation.

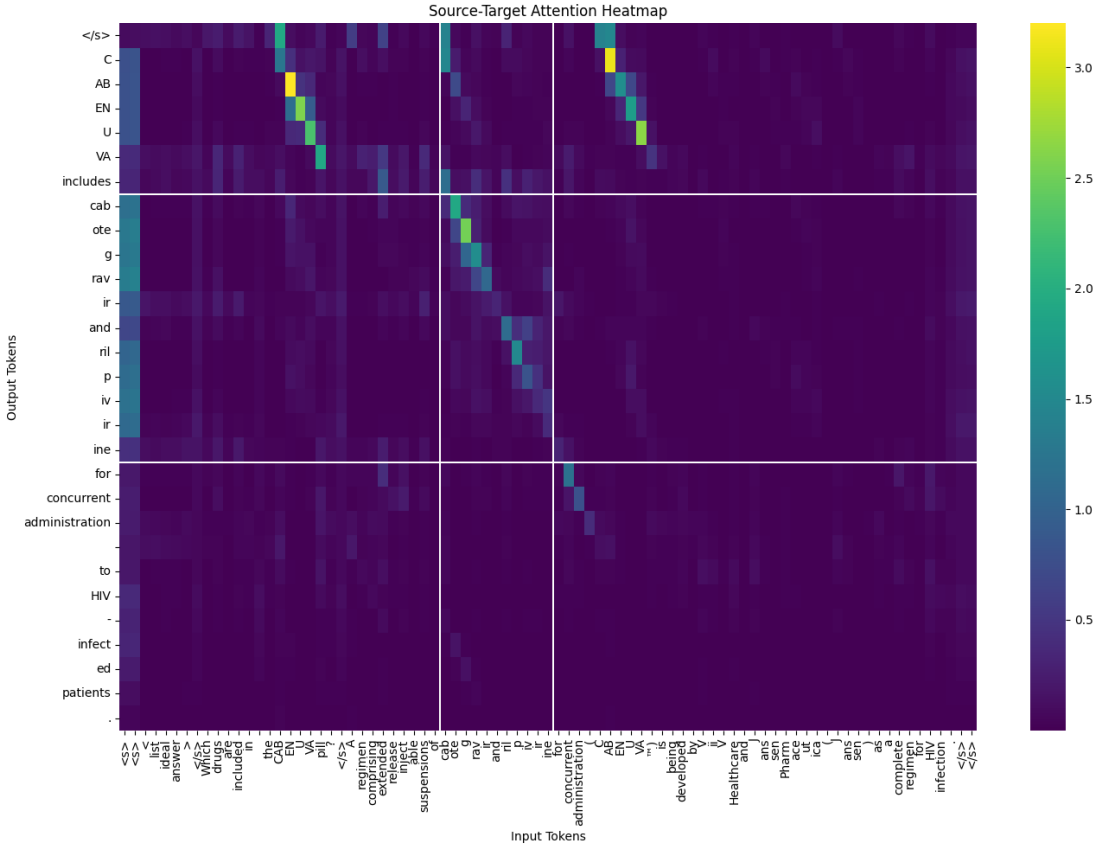

Figure 8: Cross-attention heatmaps for Pointer + KPF generation.