# Sparse Retrieval with Cross-Encoder Reranking and Heuristic Snippet Scoring for Biomedical Question Answering

BioASQ Lab at CLEF 2026

Aikaterini Maria Kouti[1], Ioannis Refanidis[1]

[1]*Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece*

**Abstract**

In this study, we employed a two-stage biomedical information retrieval pipeline for the Phase A of the BioASQ Task 14B Challenge, initially retrieving relevant documents in response to biomedical questions and then extracting the most informative snippets from the retrieved documents. The retrieval was implemented using the combination of two components: an initial retriever that uses the BM25 algorithm and BGE cross-encoder reranker in order to capture semantic relevance. For the snippet extraction, we defined the set of candidates to be all sentences extracted from the retrieved documents, as well as all the combinations of two adjacent sentences. Finally, candidate snippets were scored and ranked using the BM25 score augmented by a heuristic scoring function, which eventually led to the extraction of relevant snippets for the initial input questions.

**Keywords**

Biomedical Retrieval, Semantic Search, BM25, BGE reranker, BioASQ

## 1. Introduction

**The BioASQ Challenge.** The exponential growth of biomedical literature makes the need for efficient information retrieval (IR) mechanisms increasingly critical for healthcare professionals and researchers [1]. Since 2012, BioASQ [2] has aimed to address this by fostering research in biomedical text mining, through the orchestration of multiple challenges. Among these, the BioASQ Task-B is an annual competition focused on document and snippet retrieval and answer generation [3]. The challenge is split into two phases. During Phase A, BioASQ releases a set of biomedical questions, to which the systems need to respond with a list of documents and specific snippets of these documents that are relevant to the questions. The returned documents should be obtained from the PubMed 2026 Baseline, an annual snapshot of the PubMed biomedical literature. This collection includes approximately 41 million biomedical documents, where each document is a medical article's title and abstract, concatenated. During Phase B, BioASQ releases a set of biomedical questions accompanied with documents and snippets that are identified as relevant by medical experts. The participating systems need to respond with answers to these questions. The Challenge also includes an intermediate Phase A+ where the participants are only given the questions and are asked to return the answers directly. In this work, we address the Phase A of the Challenge by implementing a two-stage pipeline: a document retriever followed by a snippet extraction mechanism.

**Our Pipeline.** Since neural retrieval methods are computationally heavy, we rely on the BM25 algorithm, as our primary retrieval method and then rerank the top retrieved documents using the BAAI/bge-reranker-base model, a transformer-based cross-encoder that encodes the question and document to produce relevance scores. This architecture effectively leverages the BM25 lexical similarity capabilities and latency advantage, and refines the results using a context-aware model. After reranking, the 10 most relevant documents per question are retained. For snippet extraction, we segment the

*CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany*

amkouti@uom.edu.gr,aikaterinimariakouti@gmail.com (A. M. Kouti); yrefanid@uom.edu.gr (I. Refanidis)

0009-0001-1249-295X (A. M. Kouti); 0000-0003-4697-4751 (I. Refanidis)

retrieved documents into sentences and construct a list of candidate snippets. Specifically, we consider each sentence, as well as all pairs of adjacent sentences, to be candidate snippets. This design choice is motivated by the observation that a single sentence may not always be sufficient to answer a biomedical question. Indeed, in the BioASQ development dataset (described in Section 3), approximately 13.8% of relevant snippets span two adjacent sentences. We compute the relevance score of each snippet with respect to the question using BM25 and an additional heuristic. Finally, we rank the snippets in descending order, ultimately selecting the 10 most relevant to the input question.

The remainder of the paper is organized as follows. Section 2 reviews prior work on the field. Section 3 describes our proposal and its implementation details. Section 4 presents our evaluation methodology and results. Finally, Section 5 concludes our paper and discusses future work.

## 2. Related Work

**Document Retrieval.** The BM25 algorithm has been extensively adopted in prior work as the baseline for sparse lexical matching, particularly in the biomedical domain where it serves as the benchmark for evaluating newly proposed retrieval methods [4]. It is also proven to be a very latency-effective retrieval method, as it uses inverted indexes, that eliminate most of the candidates early during the search, and basic arithmetic, based on term frequency, for the computation of text similarity. Dhakal et al. [5] examined the computational efficiency of standalone BM25 in comparison to their pipeline, where they integrated dense retrieval, using SPECTER embeddings, Reciprocal Rank Fusion, and cross-encoder reranking. In terms of latency, standalone BM25 was proved to be up to 800x faster on query processing. In the biomedical field, where the literature is large, BM25 serves as a powerful retrieval tool.

However, despite its effectiveness in capturing lexical similarity, BM25 is limited by its inability to measure semantic relevance and context. This trade-off between accuracy and latency on information retrieval systems has been explored in prior work [6, 5]. To address this issue, recent studies have adopted the hybrid approach of pipelines that use BM25, followed by additional mechanisms to improve the results. This allows systems to benefit from the efficiency of sparse retrieval, while improving the accuracy of the results by capturing conceptual meaning. Zhang [7] implemented a scoring mechanism for biomedical abstracts that extends BM25 with scores of expanded terms derived from metadata, in order to retrieve biomedical articles. Their results suggest that their enhanced implementation of BM25 scoring improves the ranking of relevant documents. Luo et al.[4] proposed a retrieval mechanism where they combined BM25 with their Poly-DPR model's scoring, demonstrating that hybrid methods can surpass purely lexical approaches in biomedical retrieval.

Another common paradigm is the two-stage retriever-reranker pipeline, which uses BM25 as the primary retrieval method, followed by context-aware reranking of the retrieved documents to improve precision. This approach combines the sparse retrieval's latency advantage with improved accuracy on the results by integrating more computationally expensive models. Chaturvedula et al. [1] adopted this strategy by integrating the ms-marco-MiniLM-L-6-v2 cross-encoder reranker in their retrieval pipeline, achieving significant improvement in the precision of their results. Taken together, the existing literature illustrates the effectiveness of combining lexical and semantic methods for information retrieval tasks.

**Snippet Extraction.** A core design decision in snippet extraction is the choice of retrieval unit — that is, what constitutes a retrievable part of a document. This decision is important, as it strongly influences performance. Several approaches have been proposed in the literature. Fixed- length or sliding windows are common choices [8], but they ignore sentence boundaries. Sentence-level segmentation [9] preserves linguistic units but a single sentence may lack sufficient context. The choice depends on multiple factors, such as the available dataset, or the way the answers are distributed within the documents. In the context of the BioASQ challenge, considering each sentence as a candidate snippet has been the dominant strategy [10, 11].

After splitting the relevant documents into candidate snippets, a common approach is to assign a score to these snippets, based on how relevant they are to the given question. This scoring step serves as a mechanism to prioritize and identify the snippets that are most likely to contain the answer to the question. Various scoring methods have been explored in previous work, ranging from lexical matching techniques, such as BM25 score [11], to neural approaches based on vector representation similarity learning [12]. Recent studies also adopt a cross-encoder approach, where the query and text are concatenated and fed into a transformer model, which outputs a relevance score [13].

## 3. Methodology

Our pipeline, illustrated in Figure 1, consists of two modules that operate sequentially on each input question: a document retriever and a snippet extractor. The document retriever itself operates in two stages: it first retrieves a broad candidate set of 200 documents using BM25, then reranks these candidates using the BGE cross-encoder to produce a refined ranked list of the 10 most relevant documents. These top-10 documents are then passed directly to the snippet extractor. The following sections describe each module in detail, including corpus construction and processing.

Our system's design decisions were guided by the development dataset provided by BioASQ for Phase A, which consists of 5729 biomedical questions. Each question is provided as a JSON object that contains the question body, a unique identifier, ground truth documents and snippets.

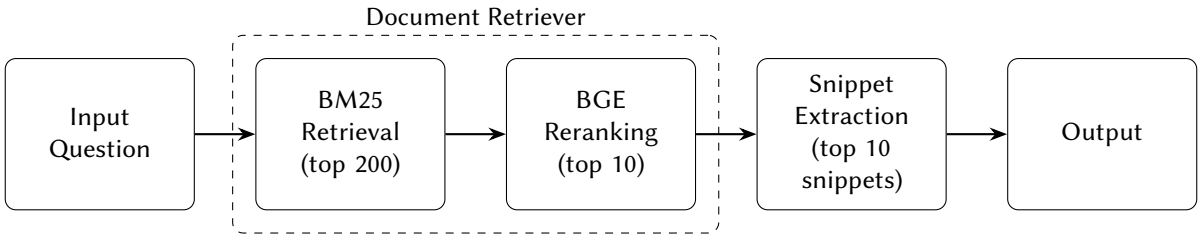


**Figure 1:** Overview of the retrieval pipeline.

### 3.1. Corpus Preparation

We parsed the PubMed 2026 Baseline XML records and extracted the PMID, title, and abstract of each article. The corpus was used in its entirety without filtering, including both MEDLINE-indexed and non-indexed PubMed records, as well as articles in multiple languages. For articles without abstracts, only the title text was retained. During parsing, XML tags and formatting artifacts were removed, and entity encodings were resolved to obtain clean textual content from the title and abstract fields.

### 3.2. Module 1: Document Retrieval

Given a biomedical question as input, our system retrieves a ranked list of the 10 most relevant documents from the PubMed 2026 Baseline corpus. We adopted a two-stage retriever-reranker architecture for our pipeline. The initial broad retrieval is performed using the BM25 algorithm. We use BM25 due to its strong effectiveness as a sparse lexical baseline and its efficiency at large-scale document retrieval, which is important given the size of the PubMed corpus. While dense retrieval methods exist, they are computationally more expensive, making BM25 a practical and robust choice for first-stage retrieval. The retrieved documents are then reranked using the BAAI/bge-reranker-base reranker, to capture semantic similarity and improve precision.

For the BM25-based retrieval stage, we constructed a Lucene inverted index offline over the corpus, using Pyserini [14], a Python toolkit for sparse retrieval indexing and search. At query time, each question is issued against the index using Pyserini's LuceneSearcher, which performs BM25 scoring. Based on this score, our system retrieves the top 200 candidate documents for the question. The

candidate pool size was selected empirically based on performance on the BioASQ development dataset, balancing recall and reranking efficiency. These 200 candidates are then passed to a neural reranker, which refines their relevance scores. We use the BAAI/bge-reranker-base cross-encoder reranker. This model operates on question–document pairs: each of the 200 candidate documents (represented as "title [SEP] abstract") is combined with the input question and scored independently. After reranking the candidate documents based on their new scores, we eventually retrieve the top-10 documents for each query.

### 3.3. Module 2: Snippet Extraction

For each input question, our system extracts the 10 most relevant snippets through a two-step process: text segmentation (in order to generate the candidate pool) and relevance scoring, using BM25 and a heuristic scoring function. These snippets are extracted exclusively from the 10 most relevant documents of the retrieval stage, not from the full corpus. An overview of this module is provided in Figure 2.

**Candidate generation.** We define two types of candidate snippets: *single* candidates, consisting of one sentence, and *pair* candidates, consisting of two adjacent sentences. Each document (title and abstract) is segmented into sentences using SciSpaCy (en_core_sci_sm), a biomedical sentence segmenter. Title sentences are added to the candidate pool directly as single candidates. For abstracts, each sentence is also added as a single candidate, but additionally, every pair of adjacent sentences $(s_i, s_{i+1})$ is added as a pair candidate. This gives our system the option to return a two-sentence snippet when a single sentence alone is insufficient to answer the query.

For each question, our system assigns a relevance score to each candidate snippet. Single and pair candidates are scored differently, as described below.

**BM25 scoring: Single Sentences.** For the computation of single sentence scores, a local BM25 index is constructed over the candidate pool, where each entry in the index is an individual snippet candidate. Each sentence is tokenized using a simple regex-based word-level tokenizer before being added to the index, as BM25 operates over discrete terms. Unlike the retrieval stage, which operates over a pre-built index of the full corpus, this index is built on-the-fly and over the candidates only. The query is tokenized using the same tokenizer, and each candidate $d$ is scored against the query $q$ using the BM25 scoring function [15], as implemented in the `rank_bm25` Python library [16]. Specifically, the scores are computed as:

$$\text{BM25}(d, q) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{(k_1 + 1) \cdot tf_{td}}{k_1 \cdot \left(1 - b + b \cdot \frac{L_d}{L_{avg}}\right) + tf_{td}} \tag{1}$$

where the IDF of term $t$ is computed as:

$$\text{IDF}(t) = \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \tag{2}$$

where $N$ is the number of candidates in the pool, $df_t$ is the number of candidates containing term $t$, $tf_{td}$ is the frequency of term $t$ in candidate $d$, $L_d$ is the length of candidate $d$, $L_{avg}$ is the average candidate length, and $k_1$ and $b$ are free parameters, set to their default values of $k_1 = 1.5$ and $b = 0.75$.

**Heuristic Integration: Sentence Pairs.** Since sentence pairs are longer, they tend to accumulate higher BM25 scores than single sentences, regardless of the informativeness of the second sentence. In order to avoid ranking pair sentences higher, when they repeat the same information, we designed a separate scoring function for such pairs. The intuition is that a pair candidate should be preferred over the individual single candidates, only if both sentences contribute distinct information, not when the high score is merely a result of repeating the same terms across both sentences. To capture this, we score

each pair based on the individual relevance score of each sentence to the question, penalized by the lexical similarity between the two sentences. This rewards pairs where both sentences are individually relevant to the query, but lexically distinct to each other. Specifically, the relevance score of a pair candidate $(s_i, s_{i+1})$ against the query $q$ is computed as:

$$\text{score}(s_i, s_{i+1}) = \frac{\text{BM25}(s_i, q) + \text{BM25}(s_{i+1}, q) - \text{sim}(s_i, s_{i+1})}{c} \tag{3}$$

where $\text{BM25}(s_i, q)$ and $\text{BM25}(s_{i+1}, q)$ are the individual BM25 scores of each sentence against the query, and $\text{sim}(s_i, s_{i+1})$ is a similarity penalty term measuring the lexical similarity between the two sentences, computed as the average of their mutual BM25 scores:

$$\text{sim}(s_i, s_{i+1}) = \frac{\text{BM25}(s_i, s_{i+1}) + \text{BM25}(s_{i+1}, s_i)}{2} \tag{4}$$

The constant $c$ is a normalization parameter tuned empirically on the development dataset, with $c = 1.8$ yielding the best performance.

Single and pair candidates are scored independently using their respective scoring functions and placed in a shared ranking pool. The top-10 candidates are then selected as the output snippets for the question.

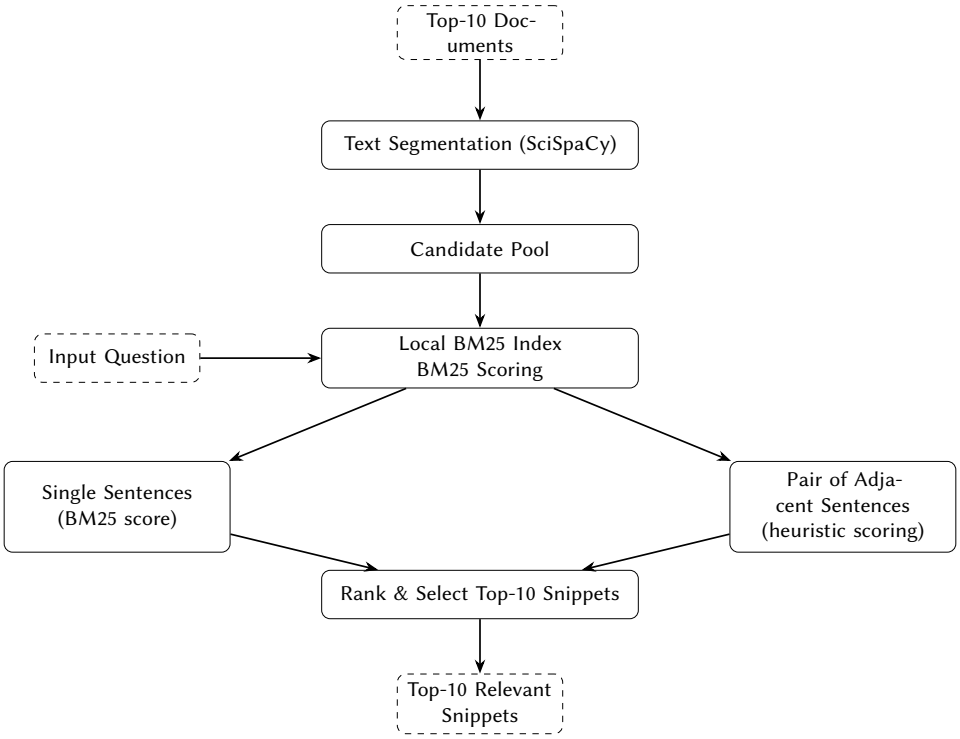


**Figure 2:** Overview of the snippet extraction module.

## 4. Results

In this section, we present our system's performance on Phase A of the BioASQ Task 14b. The challenge questions were released and evaluated in four independent batches, thus we report results for each batch separately. The official evaluation metrics used in BioASQ, include Mean Precision, Recall, F-measure, Mean Average Precision (MAP), and Geometric Mean Average Precision (GMAP), with MAP serving as the official ranking metric. Table 1 and Table 2 report the MAP and F-measure scores for document retrieval and snippet extraction respectively. For each batch, we include the best performing system,

our system, and the median across all participants. Note that the best and median F-measure values reported correspond to the F-scores of the systems ranked best and median in terms of MAP, and do not necessarily represent the best/median F-measure scores across all participants.

**Table 1**
Document Retrieval Results for BioASQ Task 14b Phase A.

| | Batch 1 | | Batch 2 | | Batch 3 | | Batch 4 | |
|---|---|---|---|---|---|---|---|---|
| **System** | **MAP** | **F-Measure** | **MAP** | **F-Measure** | **MAP** | **F-Measure** | **MAP** | **F-Measure** |
| Best | 0.2835 | 0.1686 | 0.2395 | 0.1197 | 0.2114 | 0.1178 | 0.2310 | 0.1425 |
| aida_2025_2* | 0.1949 | 0.1119 | 0.1413 | 0.0935 | 0.1521 | 0.0805 | 0.1430 | 0.0947 |
| Median | 0.1691 | 0.1112 | 0.1406 | 0.0979 | 0.1253 | 0.0964 | 0.1573 | 0.1178 |
| Rank | 19/53 | | 33/66 | | 18/62 | | 46/77 | |

* aida_2025_2 refers to our submitted system.

As shown in Table 1, our system performs above the median in Batches 1 and 3, achieving ranks of 19/53 and 18/62 respectively, demonstrating competitive document retrieval capability. In Batch 2, performance is comparable to the median (MAP 0.1413 vs. 0.1406). However, performance degrades in Batch 4 (rank 46/77, MAP 0.1430), falling below the median (0.1573). Across all batches, our system trails the best performing system by approximately 35% in MAP. This gap remains stable, suggesting a systematic ceiling, likely due to the use of purely sparse retrieval at the initial stage. Notably, our best and worst ranks are achieved in Batches 3 and 4 respectively (18/62 vs. 46/77), despite their similar absolute MAP scores (0.1521 and 0.1430). This may be explained by the increased competitiveness of Batch 4, as both the number of participating systems and the median MAP are higher on this batch.

**Table 2**
Snippet Extraction Results for BioASQ Task 14b Phase A.

| | Batch 1 | | Batch 2 | | Batch 3 | | Batch 4 | |
|---|---|---|---|---|---|---|---|---|
| **System** | **MAP** | **F-Measure** | **MAP** | **F-Measure** | **MAP** | **F-Measure** | **MAP** | **F-Measure** |
| Best | 0.1671 | 0.0914 | 0.2108 | 0.0419 | 0.1440 | 0.0780 | 0.1750 | 0.0992 |
| aida_2025_2* | 0.0769 | 0.0543 | 0.1154 | 0.0580 | 0.0489 | 0.0257 | 0.0468 | 0.0418 |
| Median | 0.0883 | 0.0549 | 0.0813 | 0.0599 | 0.0543 | 0.0260 | 0.0512 | 0.0414 |
| Rank | 22/42 | | 14/51 | | 27/50 | | 38/66 | |

* aida_2025_2 refers to our submitted system.

Table 2 presents snippet extraction results. Our system achieves its strongest performance in Batch 2 (rank 14/51, MAP 0.1154), substantially exceeding the median (0.0813). Our F-Measure score in Batch 2 (0.0580) surpasses that of the best-ranked system (0.0419), suggesting that while our system does not rank snippets as highly, it achieves better precision-recall balance. In the remaining batches, our system performs slightly below the median, with MAP scores ranging from 0.0468 to 0.0769, while F-Measure scores remain very close to the median.

An important observation is that, snippet extraction results do not consistently follow the pattern observed in document retrieval. In Batch 2, where our document retrieval rank is relatively weak (33/66), our system achieves its best snippet extraction rank, exceeding the median by approximately 42%. Conversely, Batch 3, where our document retrieval is strongest (18/62), yields our weakest snippet extraction F-Measure score (0.0257) and a below-median rank.

Overall, while document retrieval performance is competitive across most batches, snippet extraction remains a more challenging task for our system.

## 5. Conclusion

In this work, we propose a two-stage biomedical information retrieval pipeline for Phase A of the BioASQ Task 14b Challenge. For the document retrieval phase, our system combines BM25-based sparse

retrieval with BGE cross-encoder reranking, balancing BM25's speed advantage with semantic accuracy. For the snippet extraction, we introduce two types of candidate snippets: single sentences and pairs of adjacent sentences. Single sentences are scored using BM25, while pairs are evaluated using a heuristic-augmented scoring mechanism, which penalizes lexically redundant pairs. This design prevents pair candidates from ranking highly simply by accumulating BM25 scores, without contributing additional relevant information.

Across the four evaluation batches, our system demonstrates competitive performance on document retrieval, ranking above the median in three out of four batches. For snippet extraction, our system achieved its stronger result in Batch 2, surpassing the median MAP by approximately 42% and exceeding the best-ranked system's F-Measure score. However, in the remaining batches, performance was slightly below the median. This suggests that snippet extraction remains a more challenging task of our pipeline. Since we rely solely on lexical matching for snippet scoring, our system does not capture semantic relevance between the question and candidate snippets. This likely contributes to the lower performance observed in three out of four batches.

In summary, while our system demonstrates the benefits of combining lexical and semantic methods for biomedical information retrieval, there is clearly space for improvement. Future work will explore improvements on both components of the pipeline. For snippet extraction, introducing semantic relevance assessment, along with lexical matching, could address the current limitations and improve performance. For document retrieval, combining the initial BM25 stage with a dense retriever could reduce the risk of relevant documents being excluded before reranking.

## References


[1] S. Chaturvedula, Intelligent semantic search engine for biomedical literature and clinical trials: A comprehensive hybrid retrieval framework, 2026. doi:10.31224/6364.

[2] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, I. Rozhkov, E. Tutubalina, D. Dimitriadis, V. Patsiou, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of BioASQ 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, volume Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026) of *Lecture Notes in Computer Science*, Springer, 2026.

[3] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ Tasks 14b and Synergy14 in CLEF2026, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2025 Working Notes, 2026.

[4] M. Luo, A. Mitra, T. Gokhale, C. Baral, Improving biomedical information retrieval with neural retrievers, 2022. URL: https://arxiv.org/abs/2201.07745. arXiv:2201.07745.

[5] A. Dhakal, N. Chaulagain, S. S. Neupane, A comparative study of information retrieval models: Bm25 versus hybrid retrieval on the cranfield collection (2026).

[6] A. S. Sarma, P. K. Singh, Hyper-rag: Evaluating hyperparameter trade-offs in biomedical retrieval-augmented generation, in: 2025 IEEE Pune Section International Conference (PuneCon), 2025, pp. 1–7. doi:10.1109/PuneCon67554.2025.11377827.

[7] Z. Zhang, An improved bm25 algorithm for clinical decision support in precision medicine based on co-word analysis and cuckoo search, BMC Medical Informatics and Decision Making 21 (2021) 81. URL: https://doi.org/10.1186/s12911-021-01454-5. doi:10.1186/s12911-021-01454-5.

[8] Z. Wang, P. Ng, X. Ma, R. Nallapati, B. Xiang, Multi-passage bert: A globally normalized bert model for open-domain question answering, 2019. URL: https://arxiv.org/abs/1908.08167. arXiv:1908.08167.

[9] H. Wang, D. Yu, K. Sun, J. Chen, D. Yu, D. McAllester, D. Roth, Evidence sentence extraction for machine reading comprehension, 2019. URL: https://arxiv.org/abs/1902.08852. arXiv:1902.08852.

[10] D. Pappas, R. McDonald, G.-I. Brokos, I. Androutsopoulos, Aueb at bioasq 7: Document and snippet retrieval, in: P. Cellier, K. Driessens (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2020, pp. 607–623.
[11] D. Pappas, P. Stavropoulos, I. Androutsopoulos, Aueb-nlp at bioasq 8: Biomedical document and snippet retrieval, in: Proceedings of the 8th BioASQ Workshop at the Conference and Labs of the Evaluation Forum (CLEF), Thessaloniki, Greece, 2020.
[12] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://aclanthology.org/2020.emnlp-main.550/. doi:10.18653/v1/2020.emnlp-main.550.
[13] R. Nogueira, K. Cho, Passage re-ranking with bert, 2019.
[14] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations, 2021. URL: https://arxiv.org/abs/2102.10073. arXiv:2102.10073.
[15] A. Trotman, A. Puurula, B. Burgess, Improvements to bm25 and language models examined, in: Proceedings of the 19th Australasian Document Computing Symposium, ADCS '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 58–65. URL: https://doi.org/10.1145/2682862.2682863. doi:10.1145/2682862.2682863.
[16] D. Brown, rank_bm25: A collection of bm25 algorithms in python, 2022. URL: https://github.com/dorianbrown/rank_bm25.

## 6. Acknowledgments

This paper is a result of research conducted within the "MSc in Artificial Intelligence and Data Analytics" of the Department of Applied Informatics of University of Macedonia. The presentation of the paper is funded by the University of Macedonia Research Committee.

## 7. Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to assist with language refinement and grammar correction. After using this tool, the authors reviewed and edited the content as needed. All methodological decisions, experimental design, and experimental results were developed, implemented, and validated by the authors, who take full responsibility for the content of this work.