Transformer-Based Open Domain Biomedical Question Answering at BioASQ8 Challenge

Ashot Kazaryan[1,2], Uladzislau Sazanovich[1,2], and Vladislav Belyaev[1,3]

[1] JetBrains Research, Russia
{ashot.kazaryan,uladzislau.sazanovich,vladislav.belyaev}@jetbrains.com
[2] ITMO University, Russia
[3] National Research University Higher School of Economics, Russia
{287371,191872}@niuitmo.ru

Abstract. BioASQ task B focuses on biomedical information retrieval and question answering. This paper describes the participation and proposed solutions of our team. We build a system based on recent advances in the general domain as well as the approaches from previous years of the competition. We adapt a system based on a pretrained BERT for document and snippet retrieval, question answering and summarization. We describe all approaches we experimented with and show that while neural approaches do well, sometimes baseline approaches have high automatic metrics. The proposed system achieves competitive performance while being general so that it can be applied to other domains as well.

Keywords: BioASQ Challenge · Biomedical Question Answering · Open Domain Question Answering · Information Retrieval · Deep Learning

1 Introduction

BioASQ [27] is a large scale competition for biomedical research. It provides evaluation measures for various setups like semantic indexing, information retrieval and question answering, all regarding the biomedical domain. The competition takes place annually online, and each year gains more attention from research groups all around the world. The BioASQ provides necessary datasets, evaluation metrics and leaderboards for each of its sub-challenges.

More specifically, the BioASQ challenge consists of two major objectives which are called "tasks". The first is semantic indexing, which goal is to construct a search index given a set of documents, such that certain semantic relationships are held between the index terms. The second objective is passage ranking and question answering in various forms, which is given a question to return a piece of text. The returned text must either answer the question directly or contain enough information to derive the answer. In terms of the BioASQ, those objectives are called Task A and Task B, respectively.

In this work, we explore applications of the state-of-the-art model in natural language processing and deep learning in biomedical question answering. As a result, we develop a system, that is capable of providing answers in the form of documents, snippets, exact answers or abstractive text, given biomedical questions from various domains. We evaluate our system on the recent BioASQ 2020 challenge, where it achieves competitive performance.

1.1 BioASQ Tasks

Our team participated in the Task B, which involves information retrieval, question answering, summarization and more. This task uses benchmark datasets containing development and test questions, in English, along with gold standard (reference) answers constructed by a team of biomedical experts. The task is separated into two phases.

Phase A The first phase measures the ability of systems to answer biomedical questions with a list of relevant documents and snippets of text from retrieved documents. The main metric for documents and snippets is the *mean average precision (MAP)*. The average precision is defined as follows:

$$AP = \frac{\sum_{r=1}^{|L|} P(r) \cdot rel(r)}{|L_R|}$$

where $|L|$ is the number of items in a list predicted by the system, $|L_R|$ is the number of relevant items. $P(r)$ is a precision when only first r returned items are considered, and $rel(r)$ is equal to 1 if the r-th returned item is relevant. MAP and GMAP are arithmetic and geometric means of all questions in the evaluation set. For the snippets retrieval, precision is measured in terms of characters, and $rel(r)$ is equal to 1 if the returned item has non-zero overlap with at least one relevant snippet. Additional metrics are precision, recall and F1 score. A more detailed description is present in the original paper [13].

Phase B The second phase evaluates the performance of question answering, given a list of relevant documents and snippets from the previous phase. The questions are of several types: questions where the answer is either yes or no ("yes/no"), questions where the answer is a single term ("factoid"), and questions where the answer is a list of terms ("list"). Additionally, each question has an "ideal" answer, where the aim is to measure the systems' ability to generate paragraph sized passage, that answers the question.

The metrics of phase B are F1-macro for yes/no questions, *mean reciprocal rank (MRR)* [29] for factoid questions and F1 score for list questions. To evaluate answers in natural language the ROUGE [16] scores are used. We should note that human experts will additionally evaluate all systems after the contest. However, the results are not available at the time of writing this paper, thus we use only the automatic measurements to draw our conclusions.

1.2 Related Work

Most of the contemporary large scale QA systems attempt to fill the gap between massive source of knowledge and a complex neural reasoning model. Many of the popular knowledge sources are sets of unstructured or semi-structured texts, like Wikipedia [5], [33]. It is still the case for biomedical domain, where the PubMed [4] is amongst the largest sources of biomedical scientific knowledge.

During document retrieval a question answering system can benefit from structured knowledge as well. There is a rich set of different biomedical ontologies like UMLS [2] or GO [1], successful use of which is shown in different QA systems, including ones that were submitted by previous years BioASQ participants [12]. However, in our work we do not leverage such information and instead explore a more general approach, applicable to any other domain.

Many systems perform re-ranking after initial document retrieval. Specialized neural models like DRMM [8] have been successfully used in previous BioASQ challenges [3]. More recent approaches utilize transformer-based language models [28] like BERT [6] for wide variety of tasks. Applications of transformers in document re-ranking had set a new state of the art [21], including the last years' BioASQ challenges [22]. There are also systems that do document re-ranking based on snippet extraction [22], but they did not achieve the highest positions.

Some systems solve snippet extraction by utilizing methods that were originally developed for document re-ranking. [22] uses the earlier mentioned DRMM in the Task 7B and achieves top results. [23] comes up with another neural approach, employing both textual and conceptual information from question and candidate text. In our work, we experiment with different methods and show how strong baselines consistently demonstrate high metrics, given a proper document retriever.

Deep learning has shown its superiority in question answering. In the Task 5b, [30] achieve top scores by training an RNN based neural network. However, most of the modern advancements in question answering can be attributed to transformer-based models. Last years' challenges were dominated by systems that used BERT or its task-specific adapations, like BioBERT [34], [10]. In this work we experiment with a similar approach.

Deep neural and transformer based models in particular have shown their ability to tackle summarization in different setups [17], including QA summarization [15]. However, as [19] noticed, BioASQ summaries tend to look very similar to the input examples. They exploit this observation and introduce several solutions, based on sentence re-ranking, and achieve top automatic and human scores in several batches. There is also an attempt to utilize pointer-generator networks [26] for BioASQ ideal questions [7]. During the competition we extend the snippet re-ranking approach by using transformer models. Moreover, we introduce a fully generational approach, based on transformers as well.

2 Methods

In this section, we describe the system we implemented for document and snippet retrieval, as well as our question answering system. We provide the results of different approaches which we experimented with during the competition. To assess the performance of different methods more accurately, we merge all test batches of 8B task into one and use resulting 500 questions as an evaluation set. Here and during the competition, we created a development set using 100 questions from 6B and 200 questions from 7B task. Our system evolved from batch to batch, achieving its final shape in batch 5. All the ablation experiments and retrospective evaluations are performed on the system, that was used for the 5-th batch submission.

2.1 Document retrieval

For document retrieval, we implement a system conceptually similar to [21]. At first we extract a list of N document candidates using Anserini implementation of bm25 algorithm [32]. Then we use the BERT model to re-rank candidate documents and output at most ten top-scored documents.

BM25 For initial document retrieval, we used Anserini [32]. We created an index using the PubMed Baseline Repository of the 2019 year [4]. For each paper in the PubMed Baseline, we extracted the PubMed identifier, the title, and the abstract. We stored title and abstract as separate fields in the index. We applied default stopwords filtering and Porter stemming to the title and abstract provided with Anserini [31]. Overall the searcher index contains 19 million documents.

BERT Re-ranking The initial set of documents obtained with BM25 is passed to the BERT re-ranker, which assigns relevance scores to documents based on a question. We consider all documents with a score higher than a threshold to be relevant and output at most ten papers with the highest scores. To train BERT re-ranker, we created a binary classification dataset. We obtained positive examples from the gold documents of BioASQ dataset. We collected negative examples using BM25 by extracting 200 documents using a question as a query and consider all documents starting from position 100 to be non-relevant if they are not in the gold documents set. As BioASQ dataset for question answering contains questions collected from the past year contests, the relevant documents include only the papers published before the year of the competition. Usually, there are several relevant documents for the question that were published after the year of the contest. To exclude such papers from the negative examples, we calculated the maximum publication year for all relevant documents and filtered all documents published after this year from the negative examples.

Experiments We evaluated several approaches to document retrieval. First, we evaluated the performance of the BM25 algorithm, and then we applied different modifications of BERT-based re-ranker. We examined the effects of relevance score threshold as well as the number of documents obtained from the BM25 stage. The results are presented in table 1. We can see how BERT-based re-ranker consistently improves base BM25 performance.

Since our re-ranker is trained to perform logistic regression, we can vary the decision boundary to achieve an appropriate trade-off between precision and recall. However, the MAP metric, which is used as a final ranking measure, does not penalize the system for additional non-relevant documents, which means the system should always output as much documents as possible to achieve the highest score, while reducing its practical usefulness. We decided to orient our system towards both precision and recall and as a result we achieve the highest F1 scores across all batches, while maintaining competitive MAP scores.

Table 1. Results of different approaches to document retrieval on the combined test set of 500 questions from 8B. N is the number of documents returned from the BM25 stage. T is the score threshold for relevant documents.

Method	Precision	Recall	F-Measure	MAP	GMAP
BM25($N = 10$)	0.1190	0.5022	0.1730	0.3579	0.0128
BM25+BERT($N = 50$, $T = 0.5$)	**0.2892**	0.5158	**0.3334**	0.3979	0.0155
BM25+BERT($N = 50$, $T = 0.$)	0.1358	**0.5481**	0.1954	**0.4114**	**0.0221**
BM25+BERT($N = 500$, $T = 0.5$)	0.2734	0.5387	0.3249	0.4046	0.0191

2.2 Snippet Retrieval

Snippet extraction systems extract a continuous span of text from one of the relevant documents for the given question. We observe that snippets from the BioASQ training set are usually one sentence long, thus our system is designed as a sentence retriever and snippet extraction is formulated as a sentence ranking problem. We experiment with both neural and statistical approaches to tackle this challenge.

Baseline We use a simple statistical baseline for sentence ranking, which is based on measuring entity cooccurrence in question and candidate sentence. For each question and sentence we extract sets of entities Q and S respectively and compute relevance score:

$$relevance(q, s) = \frac{|Q \cap S|}{|Q|}$$

We use ScispaCy [20] `en_core_web_sm` model for extracting entitities.

Word2Vec Similarity One approach of determining sentence similarity is to map both query and candidate into the same vector space and measure the distance between them. For embedding word sequences, we use Word2Vec model, pretrained on PubMed texts [18], and compute the mean of individual word embeddings. Suppose the E_q and E_s are the embeddings of question and snippet correspondingly. The relevance of snippet for a given question is a cosine similarity between embeddings:

$$relevance(q, s) = \frac{E_q \cdot E_s}{||E_q||||E_s||}$$

BERT Similarity As the transformer pretrained on the biomedical domain should contain a lot of transferable knowledge, we check the zero-shot performance of the pretrained model. Similar to embeddings similarity, we use cosine distance between the embeddings of question and snippet. The embedding of a text span is the contextualized embedding corresponding to the special **[CLS]** token which is inserted before the tokenized text. The relevance is a cosine similarity between embeddings of question and snippet.

BERT Relevance As the task of snippet retrieval is very similar to document retrieval, we test a similar approach. We use BERT_{rel} model, trained for document ranking to assign a relevance score to the pair of question and snippet:

$$relevance(q, s) = BERT_{rel}(q, s)$$

Document Scores Finally, after assigning each question-sentence pair a relevance score, we scale the latter by additional score, based on the position of the document, which the candidate sentences are extracted from, in the list of relevant documents. Despite the simplicity of this trick, experiments show considerable improvements of evaluation metrics, which points out a strong correlation between the rank of the abstracts and the rank of the snippets from those abstracts. For each document d_i from the list of ranked relevant documents $D = d_1, d_2, \ldots, d_n$ there is a list of sentences $S_i = s_{i,1}, s_{i,2}, \ldots, s_{i,m}$ and the similarity score between query q and sentence $s_{i,j}$ is:

$$score(q, s_{i,j}) = \frac{relevance(q, s_{i,j})}{i}$$

Experiments We evaluated all described approaches to snippet retrieval. The results are presented in table 2. We can see that heuristic of adding document score into the score of a snippet allows to improve MAP scores for all approaches significantly. In line with the document retrieval, BERT relevance model has higher precision and recall with lower MAP scores. Surprisingly, retrieval based on BioBERT cosine similarly performed well even without training on any BioASQ data. We can consider the approach to be a zero-shot performance of BioBERT on the task of snippet retrieval.

Table 2. Results of different approaches to snippet retrieval on the combined test set of 500 questions from 8B. "Docs" means scaling the snippet score by the position of the source document.

Method	Precision	Recall	F-Measure	MAP	GMAP
Baseline	0.1631	0.2871	0.1841	0.6521	0.0036
Baseline + Docs	0.1733	0.2876	0.1934	0.8902	0.0020
Word2Vec Similarity	0.1702	0.2941	0.1904	0.6408	0.0054
Word2Vec Similarity + Docs	0.1727	0.2850	0.1928	**0.9350**	0.0019
BERT Similarity	0.1607	0.2621	0.1763	0.6338	0.0031
BERT Similarity + Docs	0.1733	0.2847	0.1927	**0.9374**	0.0019
BERT Relevance	**0.1931**	**0.3383**	**0.2174**	0.6926	**0.0102**
BERT Relevance + Docs	**0.1921**	**0.3344**	**0.2161**	0.8098	0.0071

2.3 Exact answers

Factoid and List questions. For factoid and list questions we generate answers with a single extractive question-answering system. Its design follows the classical transformer-based approach, described in [6]. As an underlying neural model, we use ALBERT [14] finetuned on SQuAD 2.0 [24] and BioASQ training set. SQuAD is an extractive question answering dataset, so it is well suited for BioASQ tasks. In essence, list and factoid questions can be handled by the same span extraction technique. Thus we can use the same model for both questions types, differing only at the postprocessing stage.

Throughout all the 5 batches we experiment mainly at pre- and post- processing stages, without substantial changes in the architecture of the system itself. During preprocessing, we convert input questions to the SQuAD format [25], where contexts are built from the relevant snippets, that come with each input question. The postprocessing stage is implemented in the same manner as [34]. However, for list question we additionally split the resulting extracted spans by "and/or" and "or" conjunctions, which we observed to be frequently used in chemical/gene enumerations in various biomedical abstracts. Table 3 shows the importance of this step.

Table 3. The performance of the QA model for list questions with and without splitting of answers by conjunctions as a postprocessing step. The evaluation is performed on the first batch of 8B.

Method	Mean Precision	Mean Recall	F-Measure
BioBERT	0.2750	0.2250	0.2305
BioBERT + conj split	**0.3884**	**0.5629**	**0.4315**

Yes/No questions. For yes/no questions we formulate the task as a logistic regression over question-snippet pairs and implement a transformer-based ap-

proach, similar to [34]. We use the ALBERT model and fine-tune it using SQuAD and BioASQ datasets. In the fifth batch, we additionaly use PubMedQA dataset [11] and replace the model with BioBERT. Despite that PubMedQA contains more than 200 thousand labelled examples, the average question length is twice as large as BioASQ questions' length is. We sampled 2 thousand questions with similar to BioASQ questions distribution and incorporated them into the final train set.

2.4 Summarization

Phase B also includes summarization objective, where a participating system has to generate a paragraph sized text, answering the question. We come up with different approaches for tackling this challenge.

Weak baseline BioASQ does not impose any limitations on the source of the summary. We observed that summaries tend to be one or two sentences long, reminding how snippets are composed. Straightforward approach is to use snippets, provided with the question for computing the summary. Our weak baseline selects the first snippet from the question for this purpose.

Snippet Reranking Naturally, the first snippet may not answer the question directly and clearly, despite being marked as the most relevant. A logical improvement to the baseline is to select the appropriate snippet, potentially in a question-aware manner. To make answers more granular, we split snippets by sentences and the resulting candidate pool contains snippets and snippet sentences. Sometimes, however, snippets are absent for a given question. In that case we extract the candidate sentences from the relevant abstracts. For re-ranking, we use BERT_{rel} trained for document re-ranking, as described in 2.2. Overall, we can describe this system as *sentence-level extractive summarization*.

Abstractive Summarization Our final system performs abstractive summarization over provided snippets. We use traditional encoder-decoder transformer architecture [28], where the encoder is based on BioMed-RoBERTa [9], while the decoder is trained from scratch, following BertSUM [17]. First, we pretrain the model on a summarization dataset based on PubMed, where the *target* is an arbitrary span from the abstract and the *source* is a piece of text, from which the *target* can be derived. After that, we fine-tune the model to produce summaries given the question and concatenation of relevant snippets from the BioASQ training dataset, separated with a special token.

3 Results

In this section, we present an official automatic evaluation of our system, comparing to the top competitor system. We denote our system as "PA" which

stands for the Paper Analyzer team. We additionally perform a retrospective evaluation of phase A, where the gold answers are available.

3.1 Documents Retrieval

In table 4, we present the results of our document retrieval system on all batches compared to the top competitor. The final design of our system was implemented only in the fifth batch. So, to evaluate our proposed system against our own and other participants' systems from previous batches, we computed evaluation metrics over golden answers, provided by BioASQ for the Phase B. We were able to fully reproduce official leaderboard scores for the fifth batch and show, that our final system outperforms all our previous submissions. The retrospective evaluation shows that we significantly improved our system during the contest and achieved better results with the final system.

Table 4. The performance of the document and snippet retrieval system on all batches of task 8B. "final" represents the retrospective evaluation of a system for batch 5 on previous batches. "Top Competitor" is a top-scoring submission from other teams.

		Documents			Snippets		
Batch	System	F-Measure	**MAP**	GMAP	F-Measure	**MAP**	GMAP
1	PA_{final}	0.3389	**0.3718**	0.0156	0.1951	**0.8935**	0.0019
	$PA_{batch\text{-}1}$	0.2680	0.3346	0.0078	0.1678	0.5449	0.0028
	Top Competitor	0.1748	0.3398	0.0120	0.1752	0.8575	0.0017
2	PA_{final}	0.2689	**0.3315**	0.0141	0.1487	**0.7383**	0.0008
	$PA_{batch\text{-}2}$	0.2300	0.3304	0.0185	0.1627	0.3374	0.0047
	Top Competitor	0.2205	0.3181	0.0165	0.1773	0.6821	0.0015
3	PA_{final}	0.3381	0.4303	0.0189	0.1958	0.9422	0.0028
	$PA_{batch\text{-}3}$	0.2978	0.4351	0.0143	0.1967	0.6558	0.0062
	Top Competitor	0.1932	**0.4510**	0.0187	0.2140	**1.0039**	0.0056
4	PA_{final}	0.3239	0.4049	0.0189	0.1753	0.9743	0.0015
	$PA_{batch\text{-}4}$	0.3177	0.3600	0.0163	0.1810	0.7163	0.0056
	Top Competitor	0.1967	**0.4163**	0.0204	0.2151	**1.0244**	0.0055
5	PA_{final}	0.3963	0.4825	0.0254	0.2491	**1.1267**	0.0038
	$PA_{batch\text{-}5\ (final)}$	0.3963	0.4825	0.0254	0.2491	**1.1267**	0.0038
	Top Competitor	0.1978	**0.4842**	0.0330	0.2652	1.0831	0.0086

3.2 Snippet Retrieval

In table 4, we present the results of our snippet retrieval system on all batches compared to the top competitor. Similar to the document retrieval, we performed a retrospective evaluation on all batches for the final implemented system. The evaluation shows that we significantly improved our system during the contest.

3.3 Question Answering

We submitted only baselines for batches 1 and 2, so we present results only for batches starting with 3. Overall, we achieved moderate results on the question answering task, as we mainly focused on Phase A. We believe this was caused by poor selection of the training dataset. We will analyze errors and perform additional experiments in the future. The performance of our system is presented in tables 5 and 6.

Table 5. The performance of the proposed system on the yes/no questions. "Top Competitor" is a top-scoring submission from other teams.

Batch	System	Accuracy	F1 yes	F1 no	**F1 macro**
3	ALBERT(SQuAD, BioASQ)	0.9032	0.9189	0.8800	0.8995
	Top competitor	0.9032	0.9091	0.8966	**0.9028**
4	ALBERT(SQuAD, BioASQ)	0.7308	0.7879	0.6316	0.7097
	Top competitor	0.8462	0.8571	0.8333	**0.8452**
5	BioBERT(SQuAD, BioASQ, PMQ)	0.8235	0.8333	0.8125	0.8229
	Top competitor	0.8529	0.8571	0.8485	**0.8528**

Table 6. The performance of the proposed system on the list and factoid questions. "Top Competitor" is a top-scoring submission from other teams.

Batch	System	SAcc	LAcc	**MRR**	Mean Prec.	Rec	**F-Measure**
3	PA	0.2500	0.4643	0.3137	0.5278	0.4778	0.4585
	Top Competitor	0.3214	0.5357	**0.3970**	0.7361	0.4833	**0.5229**
4	PA	0.4706	0.5588	0.5098	0.3571	0.3661	0.3030
	Top Competitor	0.5588	0.7353	**0.6284**	0.5375	0.5089	**0.4571**
5	PA	0.4375	0.6250	0.5260	0.3075	0.3214	0.3131
	Top Competitor	0.5625	0.7188	**0.6354**	0.5516	0.5972	**0.5618**

3.4 Summarization

We evaluated our systems in all the five batches. However, we were able to experiment with only one system per batch. The results are presented in the table 7. We show how simple snippet re-ranker can achieve top scores in automatic evaluation. Meanwhile the abstractive summarizer, while providing readable and coherent responses, achieves lower scores, however still very competitive ones. We hope that human evaluation will show the opposite results. We included side-by-side comparison of answers provided by both systems in the appendix (table 8).

Table 7. The performance of the proposed system on the ideal answers. "Top Competitor" is a top-scoring submission from other teams chosen by R-SU4 (F1).

Batch	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
1	Baseline	0.1118	0.1118	0.1116	0.1117
	Top competitor	0.6004	0.3660	0.6035	0.3556
2	Baseline	0.0600	0.0655	0.0615	0.0650
	Top competitor	0.5651	0.3451	0.5725	0.3376
3	Snippet Reranking	**0.5235**	0.3297	**0.5303**	0.3256
	Top competitor	0.4980	**0.3833**	0.5045	**0.3811**
4	Snippet Reranking	**0.5470**	**0.3087**	**0.5471**	**0.3001**
	Top competitor	0.5281	0.3069	0.5329	0.2987
5	Abstractive Generation	0.3696	0.3006	0.3688	0.2895
	Top competitor	**0.3867**	**0.3668**	**0.3805**	**0.3548**

4 Conclusion and Future Work

In this work, we demonstrate several approaches for our first participation in BioASQ competition. We propose a working system for each of the evaluation objectives in Task 8B, achieving top positions in several batches.

We designate a few areas for improvements. First, a more robust experimentation process is required. After the challenge, we found several errors in document and snippeet retrieval pipelines that could affect the resulting performance. Second, our system does not utilize semantic indexing, which can be implemented using different biomedical ontologies.

Appendix: Side-by-Side Comparison of Snippet Ranking and Generative Models.

Table 8. Examples of answers generated by the snippet ranking and generative question answering systems.

Question	*Sclerostin regulates what process?*
Gold	Sclerostin plays a critical role in bone homeostasis and its deficiency or pharmacological neutralization increases bone formation
Snippet Ranking	Sclerostin regulates release of bone mineral by osteocytes by induction of carbonic anhydrase 2.
Generative	Sclerostin is a locally acting regulator of late-osteoblast/preosteocyte differentiation and regulates mineralization through a local/paracrine regulator of bone metabolism rather than as an endocrine hormone. The osteocyte product sclostin, a secreted inhibitor of Wnt signaling and plays an essential role in the regulation of bone mass.
Question	*Can CD55 deficiency cause thrombosis?*
Gold	Yes, loss of CD55 is associated with thrombosis in patients with Paroxysmal nocturnal hemoglobinuria. CD55 deficiency with hyperactivation of complement, angiopathic thrombosis, and protein-losing enteropathy (the CHAPLE syndrome) is caused by abnormal complement activation due to biallelic loss-of-function mutations in CD55
Snippet Ranking	CD55 Deficiency, Early-Onset Protein-Losing Enteropathy, and Thrombosis
Generative	Yes. CD55 deficiency with hyperactivation of complement, angiopathic thrombosis, and protein-losing enteropathy (the CHAPLE syndrome) is caused by abnormal complement activation due to biallelic loss-of-function mutations in CD55
Question	*Can prevnar 13 be used in children?*
Gold	Yes, PCV13 is approved for routine vaccination of all infants as a 4-dose series at age 2, 4, 6, and 12-15 months for children who previously received 1 or more doses of the 7-valent pneumococcal conjugate vaccine (PCV7), and for children with underlying medical conditions that increase their risk for pneumococcal disease or its complications.
Snippet Ranking	PCV13 is approved for routine vaccination of all infants as a 4-dose series at age 2, 4, 6, and 12-15 months for children who previously received 1 or more doses of the 7-valent pneumococcal conjugate vaccine (PCV7), and for children with underlying medical conditions that increase their risk for pneumococcal disease or its complications.
Generative	PCV13 is approved for routine vaccination of all infants. PCV 13 is a revision of pneumococcal conjugate vaccine that should be included on pharmacy formularies.

References

[1] Michael Ashburner et al. "Gene Ontology: tool for the unification of biology". In: *Nature Genetics* 25 (2000), pp. 25–29.

[2] Olivier Bodenreider. "The Unified Medical Language System (UMLS): integrating biomedical terminology". In: *Nucleic acids research* 32 Database issue (2004), pp. D267–70.

[3] George Brokos et al. "AUEB at BioASQ 6: Document and Snippet Retrieval". In: *Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 30–39. DOI: 10.18653/v1/W18-5304. URL: https://www.aclweb.org/anthology/W18-5304.

[4] Kathi Canese and Sarah Weis. "PubMed: the bibliographic database". In: *The NCBI Handbook [Internet]. 2nd edition*. National Center for Biotechnology Information (US), 2013.

[5] Danqi Chen et al. "Reading Wikipedia to Answer Open-Domain Questions". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017). DOI: 10.18653/v1/p17-1171. URL: http://dx.doi.org/10.18653/v1/P17-1171.

[6] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: (Oct. 2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[7] Alexios Gidiotis and Grigorios Tsoumakas. "Structured Summarization of Academic Publications". In: *PKDD/ECML Workshops*. 2019.

[8] Jiafeng Guo et al. "A Deep Relevance Matching Model for Ad-hoc Retrieval". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016).

[9] Suchin Gururangan et al. "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". In: *ACL*. 2020.

[10] Stefan Hosein, Daniel Andor, and Ryan T. McDonald. "Measuring Domain Portability and ErrorPropagation in Biomedical QA". In: *PKDD/ECML Workshops*. 2019.

[11] Qiao Jin et al. "PubMedQA: A Dataset for Biomedical Research Question Answering". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019). DOI: 10.18653/v1/d19-1259. URL: http://dx.doi.org/10.18653/v1/D19-1259.

[12] Zan-Xia Jin et al. "A Multi-strategy Query Processing Approach for Biomedical Question Answering: USTB PRIR at BioASQ 2017 Task 5B". In: *BioNLP*. 2017.

[13] Martin Krallinger et al. "BioASQ at CLEF2020: Large-Scale Biomedical Semantic Indexing and Question Answering". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 550–556.

[14] Zhenzhong Lan et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: (Sept. 2019). arXiv: 1909.11942. URL: http://arxiv.org/abs/1909.11942.

[15] Mike Lewis et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *ArXiv* abs/1910.13461 (2020).

[16] Chin-Yew Lin. “Rouge: A package for automatic evaluation of summaries”. In: *Text summarization branches out*. 2004, pp. 74–81.

[17] Yang Liu and Mirella Lapata. “Text Summarization with Pretrained Encoders”. In: *EMNLP/IJCNLP*. 2019.

[18] Ryan McDonald, George Brokos, and Ion Androutsopoulos. “Deep Relevance Ranking Using Enhanced Document-Query Interactions”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018). DOI: 10.18653/v1/d18-1211. URL: http://dx.doi.org/10.18653/v1/D18-1211.

[19] Diego Mollá and Christopher Jones. “Classification Betters Regression in Query-Based Multi-document Summarisation Techniques for Question Answering”. In: *Communications in Computer and Information Science* (2020), pp. 624–635. ISSN: 1865-0937. DOI: 10.1007/978-3-030-43887-6_56. URL: http://dx.doi.org/10.1007/978-3-030-43887-6_56.

[20] Mark Neumann et al. “ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing”. In: *Proceedings of the 18th BioNLP Workshop and Shared Task*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 319–327. DOI: 10.18653/v1/W19-5034. eprint: arXiv:1902.07669. URL: https://www.aclweb.org/anthology/W19-5034.

[21] Rodrigo Nogueira and Kyunghyun Cho. “Passage Re-ranking with BERT”. In: *arXiv e-prints*, arXiv:1901.04085 (Jan. 2019), arXiv:1901.04085. arXiv: 1901.04085 [cs.IR].

[22] Dimitris Pappas et al. “AUEB at BioASQ 7: Document and Snippet Retrieval”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Peggy Cellier and Kurt Driessens. Cham: Springer International Publishing, 2020, pp. 607–623. ISBN: 978-3-030-43887-6.

[23] Mónica Pineda-Vargas et al. “A Mixed Information Source Approach for Biomedical Question Answering: MindLab at BioASQ 7B”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Peggy Cellier and Kurt Driessens. Cham: Springer International Publishing, 2020, pp. 595–606. ISBN: 978-3-030-43887-6.

[24] Pranav Rajpurkar, Robin Jia, and Percy Liang. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *arXiv e-prints*, arXiv:1806.03822 (June 2018), arXiv:1806.03822. arXiv: 1806.03822 [cs.CL].

[25] Pranav Rajpurkar et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: arXiv:1606.05250 (June 2016). arXiv: 1606.05250 [cs.CL]. URL: http://arxiv.org/abs/1606.05250.

[26] Abigail See, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017). DOI: 10.18653/v1/p17-1099. URL: http://dx.doi.org/10.18653/v1/P17-1099.

[27] George Tsatsaronis et al. "An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition". In: *BMC Bioinformatics* 16 (Apr. 2015), p. 138. DOI: 10.1186/s12859-015-0564-6.

[28] Ashish Vaswani et al. "Attention is All you Need". In: *ArXiv* abs/1706.03762 (2017).

[29] Ellen M Voorhees. "The TREC question answering track". In: *Natural Language Engineering* 7.4 (2001), p. 361.

[30] Georg Wiese, Dirk Weissenborn, and Mariana Neves. "Neural Question Answering at BioASQ 5B". In: *BioNLP 2017* (2017). DOI: 10.18653/v1/w17-2309. URL: http://dx.doi.org/10.18653/v1/W17-2309.

[31] Peter Willett. "The Porter stemming algorithm: then and now". In: *Program* (2006).

[32] Peilin Yang, Hui Fang, and Jimmy Lin. "Anserini: Reproducible Ranking Baselines Using Lucene". In: *J. Data and Information Quality* 10.4 (Oct. 2018). ISSN: 1936-1955. DOI: 10.1145/3239571. URL: https://doi.org/10.1145/3239571.

[33] Wei Yang et al. "End-to-end open-domain question answering with bertserini". In: *arXiv preprint arXiv:1902.01718* (2019).

[34] Wonjin Yoon et al. "Pre-trained Language Model for Biomedical Question Answering". In: arXiv:1909.08229 (Sept. 2019), arXiv:1909.08229. arXiv: 1909.08229 [cs.CL]. URL: http://arxiv.org/abs/1909.08229.