Comparing Statistical and Neural Question Answering in Offshore Enginnering

Vinicius Cleves de Oliveira Carmo[1], Vinicius Toquetti de Melo[1], Flávio Jaime Pol Gonçalves[1], Rodrigo da Silva Cunha[1], Ismael H. F. Santos[2], Rodrigo Augusto Barreira[2], Fabio Gagliardi Cozman[1], Edson Satoshi Gomi[1]

[1]*Escola Politécnica, Universidade de São Paulo – Cidade Universitária, São Paulo, Brazil*
[2]*CENPES, Petrobras – Rio de Janeiro, Brazil*

Abstract. This paper compares state-of-art statistical and neural models for question answering tasks in the domain of offshore engineering. We have built a document collection and a corpus of question/answer pairs for that domain, and we have used those resources to adapt and tune relevant models to the task of question answering in offshore engineering. Our experiments indicate that statistical modeling as adopted by the BM25 algorithm wins over the (neurally inspired) DPR scheme in our domain of interest.

Keywords: Transformers, Statistical language models, Information retrieval, Machine Learning.

1 Introduction

Like most engineering endeavors, offshore engineering activities depend on information often buried in extensive collections of documents. The increased production of data and information due to the emergence of digital twins and related digital technologies is bound to require even quicker information retrieval, often based on questions with semantic content rather than just keyword-based requests for data. There are now several techniques for question answering from collections of documents; a significant portion of those depend on machine learning, variously based on "statistical" or "neural" language models. For instance, the well-known BM25 algorithm for information retrieval is based on a language model where tokens are assumed independent, and probabilities are estimated from document frequency and related metrics. As a different example, the recently proposed DPR system [1] is based on language models inspired by neural networks [2]. There has been debate as to whether such neural models do offer an advantage over statistical ones, and to what extent statistical or neural models better address specific applications [3].

In that context, one is naturally led to ask whether automated question answering in offshore engineering, where answers to textual questions are found in collections of documents in that specific domain, should be dealt with by statistical or neural models. The goal of this paper is exactly to compare a state-of-art question answering scheme based on statistical modeling (more precisely, on the BM25 algorithm) and a state-of-art question answering scheme based on an architecture inspired by neural networks (more precisely, on the DPR system). To do so, we have built a document collection and a corpus containing question/answer pairs in the domain of offshore engineering, and we have tuned both approaches to that domain.

The work we report here is part of a larger effort geared towards Semantic Search in Offshore Engineering — we refer to the system we are building as SeSO. Indeed, we are putting together a question answering system that can react to semantic content, for instance, by enlarging queries and by detecting whether questions look for locations or time periods. The SeSO system consists of five modules: interface, document segmentation module, question processing module, information retrieval module, and answer processing module. In this paper, we are particularly interested in the three "middle" modules.

We review relevant literature in Section 2, and then describe SeSO in broad strokes in Section 3. In Section 4 we describe the experiments we have run to compare them. Section 5 carries an analysis of our experiments and Section 6 summarizes the results and comments on future work.

2 Background

BM25 is one of the most widely used similarity functions in information retrieval (IR) systems; in fact, it is the default similarity function in Solr and Elasticsearch, arguably the two most popular enterprise search engines. The intuition behind BM25 is that words that appear in a document more often (i.e., have a high term frequency) are more informative about the content of that document. Also, words that appear in many documents (i.e., have a low inverse document frequency) are not discriminative of the content of a particular document concerning others in the document collection. Therefore, BM25 favors documents in the collection that have many relatively rare terms from the query. Variations of the BM25 formulas can be found in the literature; here is a version adapted from Robertson and Zaragoza [4]:

$$\mathrm{BM25}_{\mathrm{score}} = \sum_i w^{\mathrm{BM25}}(q_i), \qquad \text{with} \qquad w^{\mathrm{BM25}}(q_i) = w^{\mathrm{TF}}(q_i) \cdot w^{\mathrm{IDF}}(q_i),$$

where

$$w^{\mathrm{TF}}(q_i) = \frac{tf(q_i)}{k_1\left((1-b) + b\frac{dl}{avdl}\right)} \qquad \text{and} \qquad w^{\mathrm{IDF}}(q_i) = \ln\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

where q_i is the i^{th} term in the query, $n(q_i)$ is the number of documents containing q_i, N is the total number of documents, $tf(q_i)$ is the term frequency of word q_i in the document, dl is the document length, $avdl$ is the average document length in the collection. We refer to $w^{\mathrm{BM25}}(q_i)$ as the score of the term q_i, w^{TF} and w^{IDF} as the term frequency and the inverse document frequency components of the BM25 score, respectively; k_1 and b are configurable parameters. These expressions provide no guidance on how k_1 and b should be set, but experiments have suggested that values between $0.5 < b < 0.8$ and $1.2 < k_1 < 2$ usually do well in many settings [4]. Note that, while k_1 and b can be tuned, BM25 has no learnable parameters. It depends only on the distribution of words in the corpus.

The recent progress in language modeling with deep neural networks [2] and the availability of large scale datasets, such as Natural Questions [5] has led to substantial improvements in automated Question Answering (QA). Transformer-based models such as BERT are usually trained on machine comprehension, i.e., given a question and a passage, the model learns to locate the answer in the passage. More recently, such transformer-based architectures have been applied also to retrieval tasks [1, 6, 7]. The representations learned by transformers are used to map questions and documents into a joint semantic vector space where the encodings of documents are compared to encodings of queries. Departing from the usual bag-of-words representation employed in statistical information retrieval algorithms such as BM25, transformers create representations that take into account the whole context of the sentence (due to the interaction of words in multiple layers of self-attention in encoders).

DPR is a state-of-the-art neural retrieval engine [1]. It learns to map questions and documents to a semantic vector space where documents are more similar to the questions they are relevant to. By leveraging the powerful contextual representations of BERT for sentences, DPR can retrieve documents that match the semantics of the question instead of just independent words. Specifically, for a given set of documents, DPR splits each document into 100-word passages and feeds each passage through a BERT model (passage encoder). The passage representations are obtained from the output of BERT corresponding to the special [CLS] token added at the beginning of each passage during tokenization. These vector representations are then indexed using FAISS [8], an efficient indexer for dense vectors. At query time, the question in natural language is encoded using another BERT model (query encoder), and the output is matched by dot-product similarity with the passage vectors indexed in FAISS. Figure 1 illustrates the process of indexing documents and issuing queries to DPR. Due to this semantic encoding of documents and questions, DPR has been shown to improve retrieval performance in datasets such as Natural Questions.

Despite all the advances in neurally-inspired techniques over the past years, Yang et al. [3] warn about the prevalence of weak baselines on neural information retrieval research. Their study shows that a strong BM25 with a query expansion baseline performs better than most neural models they benchmarked. Other studies have voiced doubts about neural architectures. Indeed, a primary concern regarding deep neural networks is whether they can generalize to examples not seen during training. Reddy et al. [9] experimented with zero-shot transfer learning on DPR in different datasets and compared its performance against BM25. They also proposed AugDPR, a synthetic data augmented version of DPR. They found that, for documents that are very different from the ones in Wikipedia, BM25 outperforms DPR and performs close to AugDPR.

3 Context: an overview of the SeSO system

We describe here a QA pipeline with a focus on offshore engineering, as we are implementing in our SeSO system; by doing so, we indicate the role of the information retrieval module in the context of QA systems. The

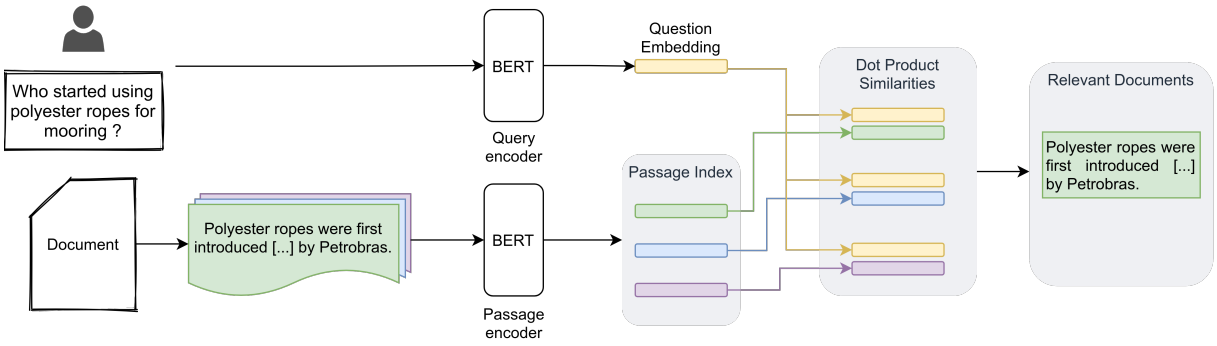

Figure 1. Indexing and quering processes in DPR.

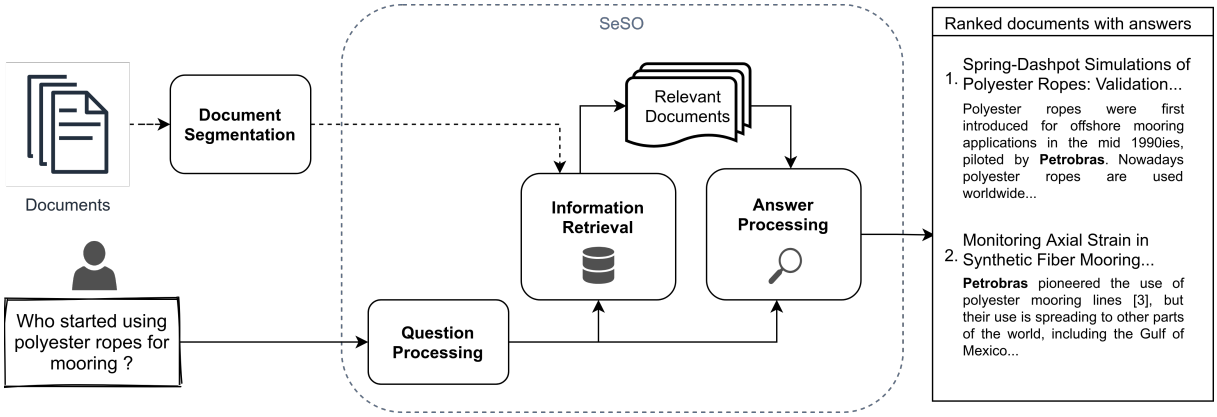

Figure 2. The SeSO system.

whole SeSO pipeline captures the best practices in QA procedures and is depicted in Figure 2. The figure shows an example question (bottom left), processed through several modules and reaching a ranked set of text fragments (right).

First, we have two components of the pipeline that control the inputs: queries and documents. They are:

Interface We have developed a website where users can ask questions in natural language and receive a list of possible answers; they can then score these answers to provide feedback and download the related source document if so desired.

Document segmentation component This component is responsible for extracting the information contained in the documents that make up our knowledge base. It extracts information from documents in different formats and lets the information retrieval module to search within well defined data structures. The original documents are in PDF format, which makes extracting information quite complicated (one faces a series of difficulties, such as the lack of a standard in the structures of documents, texts in tables, figures, graphics, and different formatting).

There are then modules that run the Question Answering steps themselves:

Question processing module The purpose of the question processing stage is to extract information from the question for the answer extraction module and to rewrite it so as to improve the quality of the following information retrieval stage. In this step, the type of question asked is detected (what, how, which, where, who), and wh-words (what, how, which, where, who) are removed from the query. While these words carry essential meaning in questions, they are hardly to be found with the same meaning in answers as the latter are usually sentences in affirmative form.

Information retrieval module The goal of the information retrieval module is to retrieve a small set of documents from the much larger document pool. The IR engine needs to have a high recall since no correct answer can be derived if the IR module recovers no answer-bearing document, but it is also desirable to have a good precision since this facilitates the task on the modules downstream. This module is our focus in this paper.

Answer processing module This module is responsible for finding the answer span among the documents retrieved by the information retrieval module. We have developed two approaches for this module: (1) the linguistic

P1	Polyester ropes were first introduced for offshore mooring applications **in the mid 1990ies**, piloted by ***Petrobras***. Nowadays polyester ropes are used worldwide, particularly for deep-water applications where catenary systems become heavy and inefficient. (...)
Q1	**When did polyester ropes start being used for mooring?**
Q2	***Who was the first company to start using polyester ropes for mooring?***

Table 1. Samples from our manually constructed question answering dataset based on OMAE papers. Questions styled according to their corresponding answers in the passage.

approach, which uses information from the question processing module along with a set of custom-designed regular expression patterns and similarity measures between query and sentence to find the answer, and (2) the neural approach, that uses the DPR Reader [1], a neural network model trained for answer selection and passage ranking. We however do not focus on this module, and these approaches, in this paper.

4 Comparing the (statistical) BM25 algorithm and the (neural) DPR system

To compare techniques and validate our implemented algorithms, we created a QA dataset of factoid question/answer pairs following the SQuAD methodology [10]: given a document, we formulate questions for which answers correspond to short text spans in the document. Questions were formulated from OMAE 2019 papers by the first and third authors, keeping in mind the kind of information a person not looking at the exact paper would be looking for. Hence we avoided asking questions about information that would be too specific in a particular paper. We kept this dataset to a limited size, with 100 question/answer pairs. Table 1 shows samples of question/answer pairs in this dataset.

We used papers from OMAE 1998-2019 conferences as documents for retrieval. Documents are judged relevant when they contain the exact string of the answer.

Our experiments focused on the question processing and the information retrieval modules. As for the answer processing module, we currently have it as an option for the end-user; we leave the study of the answer processing module to future work.

We designed three experiments in order to compare retrieval techniques:

- first, and most importantly, we compared the performance of DPR and BM25;
- second, as BM25 presented better performance than DPR, we compared how different document units (whole documents, paragraphs, or 100-word passages) affect retrieval performance;
- and finally, we studied how much distinct preprocessing steps – lowercasing, stopword removal, stemming, for example – affect performance.

We report Mean Average Precision (MAP), Mean Reciprocal Ranking (MRR), and Hit@N for all experiments. All of these are ranking metrics that range from 0 to 1, where 1 denotes a perfect score. MRR measures how close the first relevant document is to the top, while MAP measures how close all relevant documents are from the top. Hit@N measures the proportion of queries that have a relevant document among the top-N retrieved documents for the query. We calculated MAP and MRR as follows:

$$\text{MAP} = \frac{1}{Q}\sum_{i=1}^{Q}\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\frac{j}{p_j^i}\right), \qquad \text{MRR} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{p_1^i},$$

where Q is the total number of questions, m_i is the number of relevant documents returned in the i^{th} question, and p_j^i is the ranking position of j^{th} relevant document in the i^{th} question.

We now describe the experiments we run.

DPR vs BM25: This experiment aims to evaluate the performance of state-of-the-art neural and statistical retrievers. We run BM25 using Elasticsearch [1], which uses Apache Lucene as the underlying library. For DPR, we used the pre-trained model available through the Hugginface's Transformers library [11].

In order to keep a fair comparison between systems, we split documents into paragraphs and paragraphs in passages of 100 words, following the procedure in the DPR paper. We used default indexing parameters for Elasticsearch, which only lowercase text. For DPR, we experimented with document title and passage as well as empty title and passage. We report the best results in all cases.

[1] https://www.elastic.co/elasticsearch/

	Metrics					
			Hit@N			
Sistema	MAP	MRR	1	5	10	100
DPR	0.15	0.27	0.16	0.41	0.45	0.72
BM25	**0.31**	**0.48**	**0.37**	**0.60**	**0.68**	**0.87**

Table 2. DPR and BM25 performance (measured by Hit@N on the Offshore QA Dataset: BM25 does better than DPR according to all metrics.

Document units: For this experiment, we aim to evaluate how different document units (document, paragraph, 100-word passage) affect the performance of the Retriever. Larger units of text tend to perform better in our selected metrics, just because these have more words per document. So, we also report the average total words (Words@N) on retrieved documents at different numbers of documents.

We indexed whole documents, and documents split into paragraphs and paragraphs split on 100-word passages in Elasticsearch. Then we compared the performance of retrieval according to MAP, MRR, Hit@N, and Words@N.

Document and question processing: When indexing with DPR, there is not much flexibility about how to input documents into the Retriever. DPR accepts a passage and a title and builds the index. It automatically deals with word variations and lexical mismatch between queries and passages. BM25, on the other hand, requires much more tuning regarding text normalization, dealing with the lexical gap problem, and custom engineered texts pipelines to approach the query from the style of the relevant passages. Therefore, for BM25, we study how different text normalization, indexing options, and question processing affects performance. For text normalization, we evaluate how lowercasing and stemming affect the system. We also investigate stopword removal and n-grams. For question processing, we study wh-word removal.

5 Results and discussion

Table 2 shows the results for the comparison between DPR and BM25. BM25 performs significantly better than DPR on our dataset. Machine learning systems tend to degrade when applied in settings different from the training: articles from OMAE are different from Wikipedia articles, and DPR seems not to generalize well to this new set of documents. BM25 does not face such problems as it is a purely statistical retriever.

On the other hand, it should be noted that the construction of the QA dataset may favor BM25 retrieval [7]. Questions created with knowledge of the answer passage tend to have a higher overlap of words with the passages than questions created otherwise. This word co-occurrence favors BM25, which relies on this information for retrieval. In the absence of further evidence that the performance gap between DPR and BM25 is due to the limitations in dataset construction, our results indicate that BM25 is a better approach, although this comparison can be reviewed in the future should a dataset from search logs, for example, be constructed.

Table 3 shows the retrieval metrics for indexing different document units in BM25. MAP for document unit is much larger than for other units, but this comes at the cost of retrieving a much larger volume of text from the document pool. The role of the information retrieval module is to select relevant information. Presenting an excess of text can hurt downstream performance in the answer selection module and slow down the response, as the answer selection module takes some time to process each document. For the same volume of text, retrieving with paragraph or passage units can recover more information, as evidenced by the passage retrieval that recovers an average of 8,583 words as 100 passages and brings answers to 88% of questions while document retrieval recovers an average of 18,725 at 5 documents and bring answers to only 71% of questions. For that reason, we consider paragraph and passage indexing being better than document indexing, as they recover more relevant information given a limited amount of text.

Using paragraphs or passages as document units seems to be very close in effectiveness for retrieval, but passages perform slightly better.

Figure 3 shows the contribution of each processing step while indexing with BM25. We perform a grid search[2] on the processing steps, and each point is calculated based on the results with the parameter set concerning the same configuration but with the parameter unset. So, for example, from two experiments, the first with only lowercasing and MRR equals 0.48 and the second with lowercasing and wh-words removal, and MRR equals 0.50, we derive a point where wh-words removal improves MRR by 0.02. It can be noted that lowercasing continuously improves performance. Stemming also helps most of the time. These can be expected since text normalization

[2]We only condition stemming on lowercasing.

	Metrics									
			Hit@N				Words@N			
Doc. Unit	MAP	MRR	1	5	10	100	1	5	10	100
Document	0.41	0.60	0.50	0.71	0.79	0.94	3931	18725	37289	398012
Paragraph	0.31	0.48	0.36	0.61	0.66	0.88	112	506	1003	9820
Passages	0.32	0.50	0.39	0.65	0.71	0.88	89	437	865	8583

Table 3. BM25 performance with different document units. For all experiments, 100 units are retrieved for each question. Note that while metrics using the unit *Document* provide better values, it does so because of the much larger volume of text retrieved when indexing whole documents.

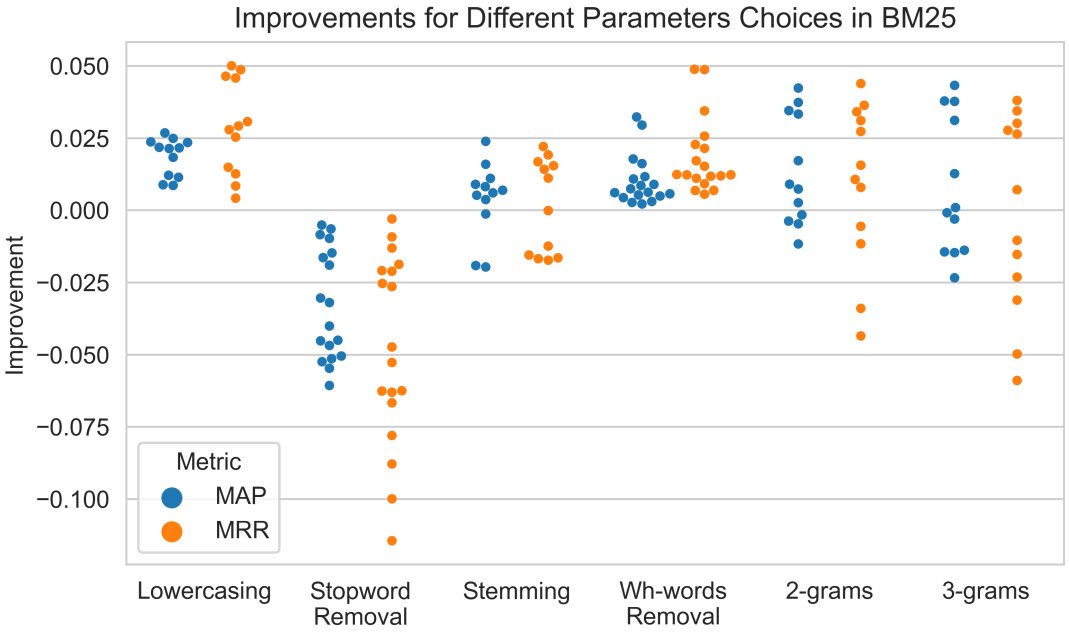

Figure 3. Improvements with respect to different parameter choices in BM25. Each point represents the retrieval improvement produced by setting the parameter in relation to the same configuration but with the parameter unset.

assists in solving the more shallow lexical gap problems. Wh-words removal is also always helpful. Wh-words are highly linked to queries, and, even if they are found in documents, they are most likely being used with another meaning, so including these words in queries only mislead the information retrieval module. 2-grams also helps most of the time, and it is also noticeable that using 3-grams does not provide further benefits to 2-grams. Stopword removal consistently hurts performance; this can be due to some of the default English stopwords used by Elasticsearch being useful for retrieval.

In short, our best configuration was found with lowercasing, stemming, 3-grams, and wh-words removal and achieved MAP=0.36, MRR=0.55, Hit@1=0.45, Hit@5=0.69, Hit@20=0.73, and Hit@100=0.90.

6 Conclusions

In this paper, we have presented a comparison between a state-of-art statistical information retrieval algorithm, BM25, and a state-of-art neurally-inspired information retrieval system, DPR, in a particular domain — Offshore Engineering. In doing so, we wanted to cut through the current hype around neural architectures and verify exactly what happens when we focus on Question Answering in a domain that is at the core of the oil and gas industry. We first described the SeSO system, our pipeline for Question Answering, to place the information retrieval module in the proper context. We then presented our experiments to define the best approach for the question processing and information retrieval modules. We also studied the effect of different document units and text processing methods.

Results indicate that BM25 performs better than DPR in our domain of interest. Experiments also show that linguistic text normalization and processing steps can increase BM25 performance, but studying their influence for each set of documents is crucial since some steps that are believed to improve performance, like stopword removal,

may in fact hurt performance in realistic scenarios.

Acknowledgements. This work was supported by ANP/PETROBRAS, Brazil (project 21721-6). Fabio G. Cozman is partially supported by Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq), grant 312180/2018-7, and acknowledges partial support from FAPESP grant 2019/07665-4.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online. Association for Computational Linguistics, 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pp. 4171–4186, 2019.

[3] W. Yang, K. Lu, P. Yang, and J. Lin. Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pp. 1129–1132, New York, NY, USA. Association for Computing Machinery, 2019.

[4] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, vol. 3, n. 4, pp. 333–389, 2009.

[5] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[6] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. REALM: Retrieval-Augmented Language Model Pre-Training. Preprint, `https://arxiv.org/abs/2002.08909`, 2020.

[7] K. Lee, M.-W. Chang, and K. Toutanova. Latent retrieval for weakly supervised open domain question answering. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 6086–6096, 2020.

[8] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, vol. , pp. 1–1, 2019.

[9] R. G. Reddy, V. Yadav, M. A. Sultan, M. Franz, V. Castelli, H. Ji, and A. Sil. Towards robust neural retrieval models with synthetic pre-training. Preprint, `https://arxiv.org/abs/2104.07800`, 2021.

[10] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 2383–2392, 2016.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, von P. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online. Association for Computational Linguistics, 2020.