Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 32, S. 682-689, Aralık 2021
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 32, pp. 682-689, December 2021
Copyright © 2021 EJOSAT
Research Article

Designing an Information Framework for Semantic Search

Alper Mıtıncık[1], İsmail Burak Parlak[2*]

[1] Galatasaray Üniversitesi, Bilgisayar Mühendisliği Bölümü,GSU-NLPLAB İstanbul, Türkiye, (ORCID: 0000-0003-3602-5237), ALPER.MITINCIK@ogr.gsu.edu.tr
[2*] Galatasaray Üniversitesi, Bilgisayar Mühendisliği Bölümü, GSU-NLPLAB İstanbul, Türkiye, (ORCID: 0000-0002-0887-4226), bparlak@gsu.edu.tr

(International Conference on Design, Research and Development (RDCONF) 2021 – 15-18 December 2021)

(**DOI:** 10.31590/ejosat.1043441)

ATIF/REFERENCE: Mıtıncık, A. & Parlak, İ.B. (2021). Designing an Information Framework for Semantic Search. *European Journal of Science and Technology*, (32), 682-689.

Abstract

New generation information retrieval procedures provide complex tools to remodel the design of search engines. Even though semantic analysis is gradually adopted by corporations, complex behavior of knowledge behind the information entails subsequent data learning models. Text models are currently in use through lexical features. Search engines with lexical methods lack contextual and semantic information. This barrier has been overcome with the development of deep learning methods. More accurate results can be retrieved by obtaining contextual information of different types of content such as text, image, video with neural models. In this study, a broad perspective of search engines was considered through lexical and semantic features. Semantic search methods were experimented then compared with lexical methods in data sets consisting of scientific documents. Since scientific documents are relatively well-formatted datasets and do not contain irrelevant content, the focus was on comparing semantic search methods and neural models throughout the study, without dealing with out-of-context data and semantic conflicts. As a result, semantic search methods performed better than lexical search. We conclude that current search-retrieval tasks require new perspectives in semantics where multimodal information is handled with deep learning strategies.

Keywords: Information retrieval, Semantic search, Deep learning, Re-ranking, Dense retrieval.

Semantik Arama İçin Bilgi Çerçevesi Tasarlanması

Öz

Yeni nesil bilgi arama prosedürleri, arama motorlarının tasarımını yeniden şekillendirmede karmaşık araçlar sağlamaktadır. Anlam tabanlı analiz profesyonel uygulamalarda kademeli olarak benimsense dahi, bilginin arkasındaki karmaşık birikimin davranışı, kademeli veri öğrenme modellerini gerektirmektedir. Metin modelleri sözlük tabanlı özelliklere dayalı olarak kullanılmaktadır. Sözlüksel yöntemlere sahip arama motorları, bağlamsal ve anlamsal bilgilerden yoksundur. Bu engel derin öğrenme yöntemlerinin geliştirilmesiyle aşılmaktadır. Metin, resim, video gibi farklı içerik türlerinin bağlamsal bilgileri sinir ağı modelleriyle elde edilerek daha doğru sonuçlara ulaşılabilir. Bu çalışmada, sözlüksel ve anlamsal özellikler üzerinden arama motorlarına geniş bir perspektiften bakılmıştır. Anlamsal arama yöntemleri denenmiş ve bilimsel dokümanlardan oluşan veri setlerinde sözlüksel yöntemlerle karşılaştırılmıştır. Bilimsel belgeler nispeten iyi biçimlendirilmiş veri kümeleri olduğundan bağlam dışı veriler ve anlamsal çatışmalarla uğraşmadan, çalışma boyunca anlamsal arama yöntemlerini ve sinir modellerini karşılaştırmaya odaklanıldı. Böylelikle, anlamsal aramanın sözcüksel aramadan daha iyi performans gösterdiği gözlenmektedir. Mevcut bilgi arama-bulma görevlerinin, çok modlu veri kümelerinin derin öğrenme stratejileriyle işlendiği anlambilimde yeni bakış açıları gerektirdiği sonucuna varılmıştır.

Anahtar Kelimeler: Bilgi çıkarımı, Semantik arama, Derin öğrenme, Tekrar sıralama, Yoğun çıkarım.

[*] Sorumlu Yazar: bparlak@gsu.edu.tr

1. Introduction

Modern information technologies are characterized through the adaptive search-retrieval tasks. The dynamics of these procedures are coupled with the relevance between the information representations and the insights of the dataset. Even though the dataset is defined with a structured data map between the documents and the queries, there are several issues during the information design. In a textual search-retrieval task, each word in a collection of documents is counted to find the connectivity which is also known as the frequency between the documents and the search query. If words in a query appear several times in a document, it is assumed that this document becomes more relevant to the query. However, word counts tend to increase if document size becomes longer. Therefore, document lengths should be computed in search-retrieval task. The optimization procedures are generally applied to generate appropriate indexes to reduce time issues. An inverted index containing the frequencies and positions of words is considered as an efficient option to perform the lexical search. Unfortunately, this mechanism does not provide contextual information of text. Lexical search might lead to retrieve irrelevant documents with polysemic properties.

In recent years, the academic scope and the corporate point of view have started to boost the number of complex approaches in semantic search-retrieval tasks. Semantic search focuses on the contextual meaning of the documents rather than the conventional lexical matching. Semantic search seeks to improve the search accuracy by understanding the content of the search query. The procedure is applied using the embeddings over all entries in the corpus into a vector space. The formalism of the query design is also embedded into the same vector space and the closest embedding. In a nutshell, three different methods are formulated for semantic search. Firstly, the sparse retrieval uses neural models where the learning function in search-retrieval lies on the token-level contextualized representations. The indexing procedure takes a long time and high storage space is the main challenge during the implementation. Secondly, the dense retrieval consists of neural model where the set of queries and document datasets are encoded in a dense vector space. Transformer based bi-encoder models are capable of encoding contexualized representations fast and efficiently. Finally, the re-ranking method leads to two stages in retrieval pipeline. The first stage is the lexical retrieval from conventional index and the latter is re-ranking the retrieved documents by using cross-encoder models. It is required high computational overhead.

Information retrieval (IR) can be applied in different media such as image, video, news, document, product. In this study, scientific documents are selected as datasets. Scientific documents are generally more structured, well-formatted for search-retrieval tasks. The Out-of-context issue is also less frequent in scientific datasets. Moreover, the information retrieval from the scientific documents can be useful in many different fields such as fact checking in the insights during the postprocessing of relevant documents through the queries.

BM25 algorithm is considered as one of the most common scoring functions in the lexical search. Robertson and Zaragoza (2009) have developed the most generic retrieval model using BM25 where the similarities and differences with other retrieval frameworks have been analyzed. They have given an overview of optimization techniques by tuning the different parameters in the models.

The attention mechanism is a promising milestone for the semantic search. Vaswani et al. (2017) have introduced a semantic information architecture where the transformer model had an eschewing recurrence to allow significantly more parallelization. This mechanism had achieved a new state in machine translation quality. Experiments on two machine translation tasks have shown these models to be superior according to semantic relevance while being more parallelizable and requiring less training time. Attention models catch on in semantic information tasks where the sequential procedures and transduction tasks are allowing to analyze query-document pairs through their dependencies on input-output similarity distances.

Furthermore, Devlin et al. (2018) have introduced a new language representation model in attention-based information analysis. Bidirectional Encoder Representations from Transformers (BERT) became one of the most pioneering models using the transformer architecture in the semantical field. BERT was originally designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model became popular in IR and natural language processing (NLP) to handle high level issues related to data semantics. Even if the computational complexity of BERT is higher, promising results have been shown recently in question answering (QA) and IR without substantial task-specific architecture modifications. In a similar way, Reimers and Gurevych (2019). have presented Sentence-BERT, a modification of the pre-trained BERT for Siamese language onto a triplet network structure. They have applied the cosine similarity to derive the semantical proximity in the sentences with the embeddings. Hofstätter et al. (2020) have proposed a cross-architecture training procedure that adapted knowledge distillation to the varying score output distributions of different BERT and non-BERT passage ranking architectures. They have evaluated the procedure of distilling knowledge from state-of-the-art concatenated BERT models to four different efficient architectures. Macdonald and Tonellotto (2020) have proposed a framework called PyTerrier which allowed a complex series of retrieval flowchart. The framework was suitable to optimize automatically the retrieval pipelines to increase their accuracy scores. Thakur et al. (2021) have introduced Benchmarking-IR, a robust and heterogeneous evaluation benchmark for the information retrieval. They have leveraged a careful selection of 18 publicly available datasets from diverse text retrieval tasks and domains and evaluated 10 state-of-the-art retrieval systems including lexical, sparse, dense, late-interaction, and re-ranking architectures on the benchmark. The dense and sparse-retrieval models have shown efficient results regarding the computational power.

In this study, we have used four scientific documents datasets from different domains. In addition, two non-scientific documents datasets are added to give an idea about the results in other textual experiments. In order to compare the results with the conventional approach, BM25 scoring is applied as a lexical search method. 16 different neural models are used, including one model in sparse retrieval, 12 models in dense retrieval, and 3 models in re-ranking. For each dataset, the total number of documents, the total number of terms, the total number of unique terms, the total number of sentences, the average number of terms per sentences, the average number of terms per documents, the total number of queries are calculated. The lexical search by using BM25 and semantic search methods such as dense retrieval, re-ranking are explained in the following section section. Embedding, attention mechanism, transformer architecture concepts are detailed in methods. All experiments are evaluated by using different metrics in the section of results section. The aim of the analysis was to reveal the promising effects of semantical IR in a comparative study. We have compared the procedures through several parameters to underline the complexity of the insights in query-document pairs. The study has been presented as follows. The second section detaisl the perspective of our methodology where the datasets, the search-retrieval tasks, the evaluation procedures and the framework have been described. The following section addresses the results in a comparative hierarchy. The best results have been highlighted in the tables. Finally, we have conluded our study by comparing lexical and semantic approaches in a broad perspective in the final section where the future steps were discussed.

2. Material and Method

2.1. Datasets

In this paper, all datasets have been fetched from the Benchmark IR repository. The corpus and query sets are enlisted for different search and retrieval purposes. In a nutshell, there are three types of data: queries, corpus, query links to relevant documents. In general, a typical query set contains id and text information. The corpus set contains id, title, text and available metadata. Query links to relevant documents register query-ID and document-ID pairs and the relevance score.

The datasets that have been used in this study were presented in Table 1. Firstly, TREC-Covid dataset was designed to build a data collection in pandemic era. The growth and pandemic information caused a big gap in search and retrieval process for medical researchers in drug design, synopsis and diagnosis levels of pandemic. The exponential growth of unstructured information during Covid-19 pandemic and subsequent lockdowns becomes the main issue of the development of a scientific data design. (Voorhees et al, 2021). Secondly, SciFact dataset was published by Allen Institute, Wadden et al (2020). It was designed as an expert-annotated dataset consisting of 1,409 scientific claims accompanied by a corpus of 5,183 abstracts. The randomly samples articles were choosen in a broad collection of medical journals with different topics for humans, animals, cell mechanisms and microbiology. Thirdly, SciDocs dataset was also created by Allen Institute, Cohan et al (2020). It was relatively larger and more diverse from other scientific datasets which allows different tasks: document classification such as Medical Subject Headings (MeSH) or Microsoft Academic Graph (MAG), citation prediction, recomendation and many others. SciDocs is consisting of many articles from different domains such as art, economics, engineering, history, medicine, and psychology. Fourthly, NFCorpus dataset was gathered from NutritionFacts.org website and was introduced by Vera Boteva et al. (2016). It is consisting of text queries linked to research articles in medical domain. Queries have been created in healthcare topics through NutritionFacts where relevance links have been extracted from PubMed using both direct and indirect links of queries. NFCorpus textual content has been extracted from titles and descriptions of videos, blog articles and Q/A and topic pages. Fifthly, ArguAna dataset was designed as the argument and the counterargument pairs extracted from the debates on idebate.org. In ArguAna study, a large corpus has been published for studying multiple counter-argument retrieval tasks. A topic-independent approach has been provided to find the best counterargument. ArguAna corpus has non-scientific and diverse documents in these domains: culture, digital freedoms, economy, education, environment, health, international, law, philosophy, politics, religion, science, society, sport (Wachsmuth et al., 2018). Finally, FiQA (Financial Opinion Mining and Question Answering) dataset has been used as non-scientific dataset. FiQA dataset is created through microblogs, reports, news (Maia et al. 2018).

At a glance, we have created multi scale dataset with 4 scientific and 2 non-scientific datasets given in Table 1. For each dataset, total number of terms, documents, sentences and unique terms, have been computed as it is shown in Table 2. Moreover, average number of terms per documents, total number of queries have been calculated.

Table 1. Numbers of Queries/Documents of Datasets

Category	Dataset	Number of Queries	Number of Documents
Scientific	TREC-Covid	50	171332
Scientific	SciFact	1109	5183
Scientific	SciDocs	1000	25657
Scientific	NFCorpus	3237	3633
Non-Scientific	ArguAna	1406	8674
Non-Scientific	FiQA	6648	57638

Table 2. Numbers of Terms of Datasets

Dataset	Terms	Unique Terms	Sentences	Terms / Sentences	Terms / Docs
Trec-Covid	16053536	185434	1126955	14.245	93.698
SciFact	658662	32591	47121	13.978	127.081
SciDocs	2686311	71106	195663	13.729	104.700
NFCorpus	504134	24388	35130	14.350	138.765
ArguAna	822246	33063	65730	12.509	94.794
FiQA	4187377	66790	419639	9.978	72.649

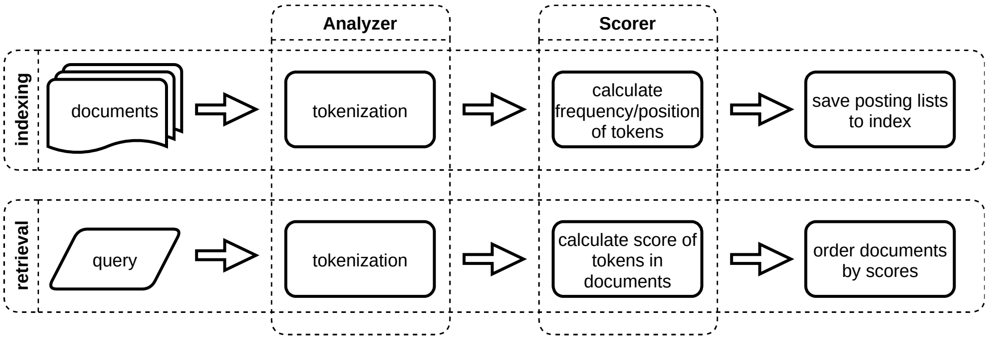
Analyzer
Scorer
indexing
documents
tokenization
calculate frequency/position of tokens
save posting lists to index
retrieval
query
tokenization
calculate score of tokens in documents
order documents by scores

Figure 1. Process of Lexical Search

2.2 Retrieval

The flowchart of IR consists of two major pipelines; the indexing and the retrieval. In the indexing step, the values such as the frequency and the position are calculated on the collected data, then converted into an easy-to-find format and saved. In the the retrieval step, the relevant documents are found by expanding and analyzing the query. There are two common search methods in which these processes are applied: lexical and semantic approaches.

Inverted index is an efficient method of information retrieval from large collections. It keeps all necessary statistics for ad-hoc information retrieval such as document frequency, term frequency per document, document length, average document length. Metadata such as name and location for document can also be saved in inverted index. There are some essential natural language processes such as tokenization, normalization, stemming, filtering stop words for each word. These procedures clean up inefficient data for retrieval. All these linguistic models are language dependent. Finally, a term list is created with a dictionary and a posting list which is called the inverted index. Frequency values and positional information are used for scoring. The most common function for score calculation is Okapi BM25 which includes TF-IDF (term frequency-inverse document frequency). Term counts for TF are calculated with logarithm function, so changes of term frequency become getting smoother. It also applies to IDF. Then, sum over all query terms, that are in index.

$$\text{TF-IDF}(d,q) = \sum_{t \in T_d \cap T_q} log\left(1 + tf_{t,d}\right) \cdot log\left(\frac{|D|}{df_t}\right) \quad (1)$$

where tf and df denotes term and document frequency respectively. D represents the total number of documents in the corpus.

BM25 is based on a probabilistic information retrieval by using TF-IDF (Robertson et al., 1994) As given the second equation, k and b determine the term frequency scaling and the document length normalization, respectively. The score of a document d given a query q which contains the words $\mathrm{q}_1, \dots, \mathrm{q}_\mathrm{n}$ is given by:

$$\text{BM25}(d,q) = \sum_{i=1}^{n} \frac{IDF(q_i) \cdot TF(q_i,d) \cdot (k_1 + 1)}{TF(q_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)} \quad (2)$$

As shown in Figure 1, in the indexing step, all documents are analyzed as language-dependent or language-independent, and terms are retrieved. Terms are counted, average document lengths are calculated for use in BM25 function, documents are counted. Posting lists are created with terms and position data. All values are saved in the lexical index. In the retrieval step, the query is analyzed similarly. BM25 scores are calculated using the terms in the query and all documents containing the terms, and the documents are ranked accordingly.

On the other hand, the semantic information retrieval considers the contextual basis. Neural language models are capable of finding contextual information. The embeddings are generated for both queries and documents via complex language models. The similarity between these embeddings is calculated and ranked. Finding more relevant document for query depends on the semantic understanding mechanism of the query and the documents. The vector representation describes an embedding which are determined using different levels such as word embedding, char n-gram embedding, sentence embedding, document embedding etc. Different neural models use different levels of input representation and learning representation. Word embedding is commonly used. In word embeddings, vector representations of words are provided (Bojanowski et al., 2016, Mikolov et al., 2016) Usually distance between two embeddings is calculated, especially cosine similarity is widely used. The closest neighbors represent the semantically relevant documents.

Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are commonly used in information retrieval. CNN can be used for n-gram modeling and character embedding. RNN characterizes the global patterns in sequence data by operating words in corpus through sequences. RNN is widely used in text data. Word representations are encoded in RNN layers and decoded in other RNN layers. Output is created with functions such as softmax probability via vocabulary. The fixed-length context vector can be a bottleneck. Attention mechanism helps to solve this problem. It allows to find relevant parts of the input. It creates a weighted average context vector. Weights are based on a softmax, sum up to 1. Attention is parameterized and trained end-to-end with the model. The input involves queries and keys of dimension d_v, and values of dimension d_v. The dot products of the query with all keys are computed, these are divided each by $\sqrt{\mathrm{d}_\mathrm{k}}$, and applied a softmax function to obtain the weights on the values. The attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. The matrix of outputs is computed below.

$$\text{attention}(Q,K,V) = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (3)$$

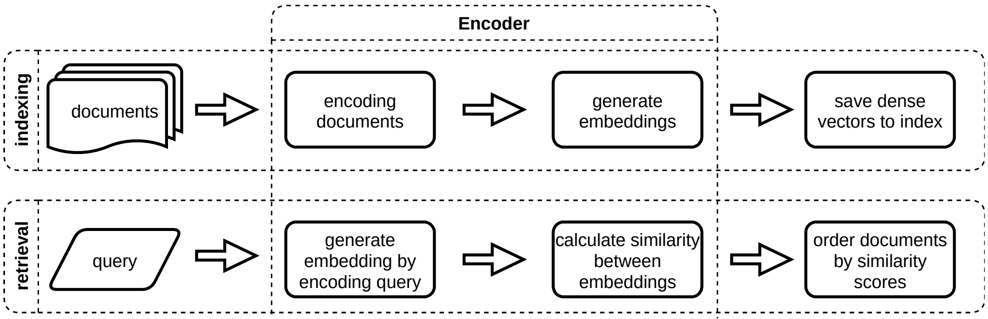
Encoder
indexing
documents
encoding documents
generate embeddings
save dense vectors to index
retrieval
query
generate embedding by encoding query
calculate similarity between embeddings
order documents by similarity scores

Figure 2. Process of Semantic Search

The global mechanism of semantic search is given in Figure 2. For a tensor, the attention is calculated with a series of matrix multiplications over the sequence. This is an efficient way computationally. It operates on sequences of vectors. Transformers are stacked with multiple layers. In each transformer layer, each vector are projected with 3 linear layer to Query, Key, Value. These projections are transformed to another multi-head dimension. Query and Key matrices are multiplied. Query and Key attention is calculated via softmax.

BERT stands for Bidirectional Encoder Representations from Transformers. BERT has wordpiece tokenization and embedding. It covers infrequest terms in small vocabulary. It has many dimensions and layers, base version has 12 layers and 768 dimensions. BERT adds trained position embeddings and sequence embeddings. Token embeddings can be word pieces. Bi-encoders perform self-attention over the query and candidate document separately, map them to a dense vector space, and then combine them at the end for a final representation. Therefore, bi-encoders are able to index the encoded candidates and compare these representations for each input resulting in fast prediction times. Cross-encoders perform cross self-attention over a given input and candidate document and tend to attain much higher accuracies than their counterparts. Bi-encoders method usually achieves lower performance compared with cross-encoders method and requires a large amount of training data. Therefore, bi-encoders are used in dense retrieval for all documents, cross-encoders are used in re-ranking with candidate documents coming from first stage retrieval.

2.3 Evaluation

Feedbacks and labeled datasets can be used to improve information retrieval system. With these data, learning-to-rank methods are applied in lexical search, and models are retrained in semantic search. Evaluation metrics are required to implement these methods. Relevance of documents in resultset can be binary or graded. Evaluation metric should be chosen accordingly. Basic evaluation metrics precision and recall can be used. However, in information retrieval systems, only the number of relevant documents in resultset is not sufficient. It is also important to have relevant documents at the top. Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) are more commonly used, where ranking is involved in calculation. NDCG supports graded relevance while MAP uses binary relevance values.

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances. Recall (also called sensitivity) is the fraction of relevant instances that were retrieved. Precision is equal to the number of related documents retrieved divided by the number of retrieved documents. Recall is equal to the number of retrieved related documents divided by the total number of related documents.

MAP for a set of queries is the mean of average precision scores for each query. In this calculation, it becomes important to have relevant documents at the top. @k notation is used for top k results.

$$\mathrm{MAP(Q)} = \frac{1}{|Q|} \cdot \sum_{q \in Q} \frac{\sum_{i=1}^{k} P(q)_{\#i} \cdot rel(q)_i}{|rel(q)|} \qquad (4)$$

Q is query set, $\mathrm{P(q)_{\#i}}$ equals precision of query q after first i documents. $r\mathrm{el(q)_i}$ equals binary relevance of document at position i. $|\mathrm{Q}|$ and $|\mathrm{rel(q)}|$ are shows numbers of queries and relevant documents. MAP is the mean of average precision over all the queries.

Normalized Discounted Cumulative Gain (NDCG) allows to use multilevel grading in the evaluation. Grades of documents are divided into order of documents and DCG is calculated with their sum. Then DCG values in all queries are divided by sorted DCG values. Usually four levels of perfectly relevant, highly relevant, relevant, irrelevant grades are used.

$$\mathrm{DCG(Q)} = \sum_{d \in D, i=1} \frac{rel(d)}{log_2(i+1)} \qquad (5)$$

$$\mathrm{NDCG(Q)} = \frac{1}{|Q|} \cdot \sum_{q \in Q} \frac{DCG(q)}{DCG\left(sorted\big(rel(q)\big)\right)} \qquad (6)$$

Q and D are query and its result document sets, $|\mathrm{Q}|$ shows number of queries. $\mathrm{rel(d)}$ equals relevance grade for single query-document pair, and $\mathrm{rel(q)}$ is list of all relevance grades for a query. $\mathrm{sorted\big(rel(q)\big)}$ function returns graded documents by descending relevance.

Natural Language Toolkit (NLTK) was used in dataset analysis. Elasticsearch was used in lexical search experiments. Benchmark-IR was used in semantic search experiments. Trec-Covid dataset is provided under publisher-specific licenses, declared in the paper. SciFact is provided under the CC BY-NC 2.0 license. SCIDOCS is provided under the GNU General Public License v3.0 license. NFCorpus, FiQA do not report the dataset license in the paper or a repository. ArguAna is provided under the CC BY 4.0 license. Neural models are derived through Hugging Face (https://huggingface.co). We have adapted the neural models using Python programming language

3. Results

3.1. Evaluation Metrics

In order to analyze our approaches, a document collection, query collection and a set of relevance judgments are prepared. Trec-Covid and NFCorpus, have 3-level graded relevance judgments. SciFact, SciDocs, and non-scientific datasets have binary relevance judgments.

In a search engine, users want to see related documents on the top of results. Therefore, ranking is as important as relevance. Even though precision and recall metrics are widely used in other domains, they are not effective in information retrieval. Instead, MAP and NDCG metrics are preferred, which also use rankings in the result documents.

Table 3. Evaluation Scores of Top k Results

Metric	**Top 10**	**Top 100**	**Top 1000**
Precision	0.556	0.433	0.199
Recall	0.014	0.105	0.424
MAP	0.012	0.070	0.197
NDCG	0.513	0.417	0.432

For the Trec-Covid dataset, evaluation results are given for the top 10-100-1000 results obtained using the all-mpnet-base-v2 model in dense retrieval method in Table 3.

Unlike precision and recall, MAP and NDCG do not depend on the number of retrieved results. Regardless of the number of results, the relevancy or accuracy value increases if the relevant documents are on the top.

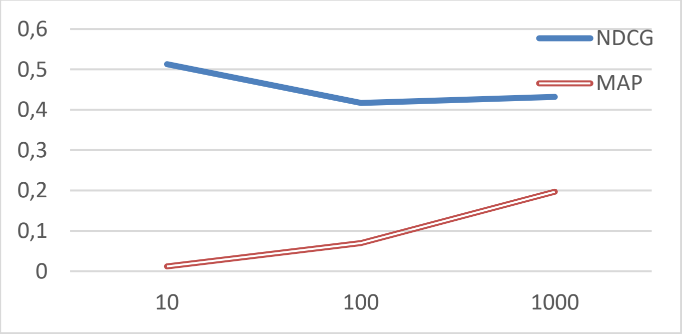

Figure 3. NDCG and MAP metrics

NDCG is calculated with graded relevance judgments. Therefore, it is an accurate and fair evaluation metric if more relevant results are at the top. For example, in the results of the Trec-Covid dataset, it can be seen that more relevant documents are not on the top in the top 100 results, while more relevant results are on top for the Top 10 results as given in Figure 3. MAP, on the other hand, shows the top 100 results as better than the top 10 results because it does not use grade.

3.2 Experiments

Variety of datasets in which the models are trained prevents model from being biased. Having a large corpus of datasets also helps to reduce the problem of out of vocabulary. Training datasets contain query and document tuples. In the experiments, models are elected with 4 different dataset groups: ALL, MULTI-QA, MS-MARCO, Specter.

- Specter dataset contains scientific texts. It has about 684K training tuples.
- MS-MARCO (Microsoft Machine Reading Comprehension) is a large-scale dataset focused on machine reading comprehension, QA, and passage ranking (Bajaj et al., 2016). There are different sizes of dataset. Small version is used in selected models. It has 532,761 training tuples.
- MULTI-QA is collected for QA purpose. It is transformed as suitable for IR tasks. It contains many different QA datasets such as WikiAnswers, Stack Exchange, MS-Marco, Amazon QA, Yahoo Answers etc. It has 214,988,242 training tuples.
- ALL contains both QA and IR datasets such as Reddit comments, PAQ, S2ORC, Code Search, COCO Image Captions, Specter, SearchQA, Flickr Wikipedia, SQuAD etc. It has 1,170,060,424 training tuples.

Pooling operation allows to derive a fixed sized sentence embedding. There are three main pooling strategies: Using the output of the CLS-token, computing the mean of all output vectors (MEAN-strategy), and computing a max-over-time of the output vectors (MAX-strategy). For a sentence 5 tokens long with CLS-token, each token in the input sentence is embedded in a tensor and is represented in a vector space. After the pooling operation, sequence dimension is squashed and it represents a pooled embedding of the input sequence.

6 base models are used in the experiments: MPNet, MiniLM, BERT, DistilBERT, RoBERTa, ELECTRA.

- BERT is a bidirectional transformer pretrained using a combination of masked language modeling objective and next sentence prediction on a large corpus. DistilBERT is a faster Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT’s performances (Sanh et al., 2019). RoBERTa was built on BERT. It modifies key hyperparameters, removing the next-sentence pretraining objective (Liu et al., 2019).
- MPNet adopts a novel pre-training method, named masked and permuted language modeling, to inherit the advantages of masked language (Song et al., 2020).
- MiniLM has effective approach to compress large Transformer based pre-trained models, termed as deep self-attention distillation. (Wang et al., 2020).
- ELECTRA is a new pretraining approach which trains two transformer models: the generator and the discriminator. The generator is replacing tokens in a sequence, and is therefore trained as a masked language model. The discriminator is identifying which tokens were replaced by the generator in the sequence (Clark et al, 2020).

Table 4. NDCG@10 Scores for Neural Models

Method	Training	Model	Pooling	Scoring	Scientific				Others	
					Trec-Covid	SciFact	SciDocs	NFCorpus	ArguAna	FiQA
Dense	ALL	**mpnet-base-v2**	mean	dot	0.513	**0.655**	**0.237**	**0.332**	0.465	**0.499**
		distilroberta-v1	mean	dot	0.528	0.631	0.216	0.292	0.479	0.394
		MiniLM-L12-v2	mean	dot	0.508	0.626	0.218	0.322	0.471	0.372
	MULTI-QA	mpnet-base-dot-v1	cls	dot	0.618	0.589	0.166	0.318	0.503	0.487
		MiniLM-L6-cos-v1	mean	cos	0.558	0.540	0.157	0.296	0.491	0.363
		distilbert-dot-v1	cls	dot	0.693	0.548	0.157	0.313	0.410	0.432
		distilbert-cos-v1	mean	cos	0.563	0.595	0.160	0.302	**0.510**	0.400
	MSMARCO	distilbert-base-tas-b	cls	dot	0.481	0.642	0.148	0.318	0.427	0.300
		distilbert-dot-v5	mean	dot	0.663	0.594	0.140	0.298	0.348	0.286
		bert-co-condensor	cls	dot	**0.726**	0.600	0.139	0.318	0.379	0.285
		roberta-base-ance-firstp	cls	dot	0.653	0.511	0.121	0.235	0.418	0.294
	Specter	allenai-specter	cls	dot	0.358	0.506	0.142	0.185	0.320	0.061
Sparse	MSMARCO	distilbert-base-v1	mean	cos	0.606	0.666	0.058	0.353	0.352	0.355
Re-ranking	MSMARCO	**MiniLM-L-6-v2**	-	cos	**0.757**	**0.686**	**0.165**	**0.365**	0.416	0.384
		TinyBERT-L-2-v2	-	cos	0.728	0.663	0.152	0.352	**0.418**	0.333
		electra-base	-	cos	0.697	0.673	0.153	0.344	0.400	**0.386**

Table 4 contains the results of all semantic search methods in the NDCG@10 evaluation metric. Neural models are grouped by training datasets. Pooling strategy and similarity function are specified for each neural model. Sparse retrieval does not seem efficient in production due to disk size and computation cost. The models that performed better in re-ranking and dense retrieval methods are marked in bold. ms-marco-MiniLM-L-6-v2 as cross-encoder and all-mpnet-base-v2 as bi-encoder are selected.

Table 5. Lexical vs. Re-ranking vs. Dense Retrieval

Dataset	Lexical	Re-ranking	Dense Retrieval
	BM25	*BM25 + Cross-Encoder*	*Bi-Encoder*
Trec-Covid	0.688	**0.757**	0.513
SciFact	0.685	**0.686**	0.655
SciDocs	0.164	0.165	**0.237**
NFCorpus	0.343	**0.365**	0.332
ArguAna	**0.471**	0.416	0.465
FiQA	0.253	0.348	**0.499**

As a result, ranking success increased in all datasets in the re-ranking method. In 4 scientific datasets, results are increased 10%, 1‰, 6‰, and 6% respectively as given in Table 5. 4% on average. In dense retrieval, 3 out of 4 results have decreased, unfortunately. It has better results in only one scientific dataset. Among the models used in dense retrieval, pooling and scoring are compared in the same base models. Mean pooling has performed better on 3 of 4 scientific datasets. Cosine similarity performed better for the same base models. Looking at the training datasets, the models trained with more and diverse datasets performed better by far. For base models, MPNet and MiniLM seem more successful than others.

4. Conclusion

In current information search-retrieval tasks, lexical search is mostly preferred due to its time complexity, hardware costs and the performance of relevance scores between query-document pairs. The main challenge in a semantical search-retrieval tasks is to prepare a benchmark dataset in a broad perspective. Since scientific documents are well-formatted, data-related problems have been largely resolved. In this way, we have focused on the comparative analysis of semantic search methods with different neural models. As a result, it was seen that semantic search has shown better results than lexical search. Neural models were differentiated into training dataset groups, base models, pooling strategy, and similarity function. The sparse retrieval method was not efficient due to disk size requirements and computational cost. Models that perform better in re-ranking and dense retrieval methods were selected and compared with lexical search. In the re-ranking method, the ranking accuracy of all data sets increased by an average of 4 percent. In dense retrieval, the accuracy rate decreased in 3 out of 4 data sets. As a result, it is seen that semantic search performs better than lexical search in the re-ranking method. Even if we note that our results are promising, there are shortcomings in the study. The results could be examined by grouping scientific documents by domain-specific or general. By detailing the data analysis, the data-centric success of the methods and models could be determined. In the next stages, the neural model comparison results can be used in model pre-training and model-tuning studies for IR purposes. It can be a guide for those who want to use semantic search in production. Our future steps of semantic search-retrieval tasks will focus on the creation of a new generalized non scientific dataset where the current lexical and semantical methods will be dealt with

Furthermore, a multilingual framework will be also added to the current scheme.

5. Acknowledgements

All experiments are publicly available at https://github.com/alpers/semantic-search

References

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T. (2016). MS MARCO: A Human Generated MAchine Reading COmprehension Dataset.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2016). Enriching Word Vectors with Subword Information.

Boteva, V., Gholipour, D., Sokolov, A., & Riezler, S. (2016). A full-text learning to rank dataset for medical information retrieval. Lecture Notes in Computer Science, 716-722. doi:10.1007/978-3-319-30671-1_58

Clark, K., Luong, M., Le, Q., Manning, C. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.

Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D. (2020). SPECTER: Document-level Representation Learning using Citation-informed Transformers.

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A. (2020). Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Macdonald, C., & Tonellotto, N. (2020). Declarative Experimentation in Information Retrieval using PyTerrier. Proceedings Of The 2020 ACM SIGIR On International Conference On Theory Of Information Retrieval. doi: 10.1145/3409256.3409829

Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., & Balahur, A. (2018). WWW'18 Open Challenge. Companion Of The The Web Conference 2018 On The Web Conference 2018 - WWW '18. doi: 10.1145/3184558.3192301

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2016). Efficient Estimation of Word Representations in Vector Space.

Reimers, N., Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.

Robertson, S., Walker, S., & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. Information Processing & Management, 36(1), 95-108. doi:10.1016/s0306-4573(99)00046-1

Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations And Trends® In Information Retrieval, 3(4), 333-389. doi: 10.1561/1500000019

Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Song, K., Tan, X., Qin, T., Lu, J., Liu, T. (2020). MPNet: Masked and Permuted Pre-training for Language Understanding.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I. (2021). BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I. (2017, June 12). Attention Is All You Need.

Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W., Lo, K., Roberts, K., Soboroff, I., Wang, L. (2021). TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection.

Wachsmuth, H., Syed, S., & Stein, B. (2018). Retrieval of the best counterargument without prior topic knowledge. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). doi:10.18653/v1/p18-1023

Wadden, D., Lin, S., Lo, K., Wang, L., Zuylen, M., Cohan, A., Hajishirzi, H. (2020). Fact or Fiction: Verifying Scientific Claims. Retrieved November 28, 2021, from the arXiv database.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers.