Information Retrieval Models for Small Domain Specific Datasets

Tanay Trivedi
Department of Computer Science
New York University
tt1087@nyu.edu

Shuvadeep Saha
Deptartment of Mathematics
New York University
ss15592@nyu.edu

Abstract

Information retrieval is a crucial task in many domains, enabling users to efficiently access relevant information from vast amounts of data. However, traditional retrieval models often struggle with small domain-specific datasets, which are characterized by limited vocabulary and complex domain-specific terminology. In recent years, several approaches to information retrieval have emerged, including BM25, the word2vec based Dual Embedding Space Model, and the transformer based coCondenser. This paper aims to explore and compare these techniques for information retrieval on small domain-specific datasets, examining their effectiveness in handling domain-specific terminology and improving retrieval performance. By providing a comprehensive analysis of these approaches, this paper will offer insights into their strengths and limitations, and inform future research in this field.

1 Introduction

Information retrieval is an essential component of many modern applications that deal with large volumes of unstructured textual data. The goal of information retrieval is to retrieve relevant information from a corpus of documents in response to user queries. Over the years, several approaches have been developed to address this problem, including vector space models, probabilistic models, and deep learning-based models. In recent years, techniques such as BM25, word2vec, and transformers have gained popularity in the field of information retrieval due to their ability to capture complex semantic relationships between words and documents. In this paper, we will explore these three techniques and evaluate their effectiveness in retrieving relevant documents from a large corpus of textual data. Specifically, we will examine how these techniques can be used to improve the accuracy and efficiency of information retrieval systems.

2 Approach

The Continuous Bag-Of-Words (CBOW) model acquires the embedding of a word by maximizing the logarithm of the conditional probability of the word, given the context words found within a predetermined window surrounding that word. In other words, the model takes the words in the context window as input and tries to predict the central (missing) word. When utilizing Word2Vec, an essential aspect that is frequently disregarded is the existence of two distinct sets of vectors (represented as c and w in the text and henceforth labeled as the IN and OUT embedding spaces). These two sets correspond to the W_{IN} and W_{OUT} weight matrices displayed in Figure 1.This is rightfully pointed out by Mitra et al. (2016) in their work.

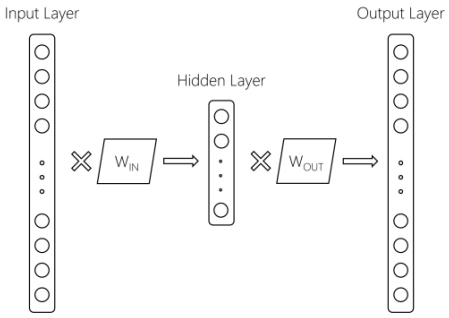

Figure 1: The architecture of a word2vec (CBOW) model considering a single context word. W_{IN} and W_{OUT} are the two weight matrices learnt during training and corresponds to the IN and the OUT word embedding spaces of the model. (Mitra et al., 2016)

To quote Mitra et al. (2016) "A key challenge for term-matching based retrieval is to distinguish whether a document merely references a term or is about that entity." Even though a query term may occur an equal number of times in two distinct documents, it is possible that only one of the documents is genuinely about that entity and therefore pertinent to the query, while the other just

mentions the term and may not be relevant to the query.However, these passages would be indistinguishable under term counting. The semantic similarity of non-matched terms (i.e. the words a TF feature would overlook) are crucial for inferring a document's topic of focus– that is its **aboutness**. CBOW is an appropriate choice for modeling a document's aboutness because it can capture word co-occurrence through missing word prediction. The embedding spaces that the model learns are rich in information about the distributional characteristics of words. This motivated the Dual Embedding Space Model (DESM) by Mitra et al. (2016).

As mentioned earlier, the CBOW model comprises two distinct embedding spaces (IN and OUT), and their interactions capture extra distributional semantics of words that cannot be deduced by examining either of the two embedding spaces individually.The CBOW model brings the IN vectors of words nearer to the OUT vectors of other words that they frequently co-occur with, causing words that appear in comparable contexts to be positioned closer to one another within both the IN and OUT embedding spaces. As a result, words that are similar in type or function exhibit higher IN-IN (or OUT-OUT) cosine similarities, while those that frequently co-occur in the training corpus display higher IN-OUT cosine similarities, indicating topical similarity. This results in at least two variations of the DESM, each corresponding to retrieval in either the IN-OUT space or the IN-IN space.

Given query q_i and document d_j as the embedding vectors for the ith and the jth term of the query and the document, respectively, we define the Dual Embedding Space Model as:

$$DESM_{IN-OUT}(Q, D) = \frac{1}{|Q|} \sum_{q_i \in Q} \frac{q_{IN,i}^T D_{OUT}}{\|q_{IN,i}\| \, \|D_{OUT}\|}$$

$$DESM_{IN-IN}(Q, D) = \frac{1}{|Q|} \sum_{q_i \in Q} \frac{q_{IN,i}^T D_{IN}}{\|q_{IN,i}\| \, \|D_{IN}\|}$$

where,

$$D = \frac{1}{|D|} \sum_{d_j \in D} \frac{d_j}{\|d_j\|}$$

D is the centroid of all the normalized document word vectors serving as a single embedding for the whole document. Note that taking the centroid of the document word vectors is equivalent to computing the similarity between all query-document word pairs (Mitra et al., 2016).

3 Experiments

3.1 Datasets

We considered two major public datasets 1) NFCorpus (Lee et al., 2019); and 2) TREC-COVID (Wang et al., 2020).

The NFCorpus (Lee et al., 2019)is a full-text English retrieval data set for Medical Information Retrieval. It contains a total of 3,244 natural language queries (written in non-technical English, harvested from the NutritionFacts.org site) with 169,756 automatically extracted relevance judgments for 9,964 medical documents (written in a complex terminology-heavy language), mostly from PubMed. It consists of 9k training, testing and development samples.

The TREC-COVID (Wang et al., 2020) dataset is a collection of scientific articles related to the COVID-19 pandemic, compiled by the National Institute of Standards and Technology (NIST) as part of the Text Retrieval Conference (TREC) COVID-19 Challenge. The dataset includes over 60,000 articles from various sources, including PubMed, bioRxiv, medRxiv, and other preprint servers. The goal of the challenge is to develop effective methods for retrieving relevant information from the large and rapidly growing corpus of COVID-19 literature. It consists of 180k training, testing and development samples.

3.2 Metrics

The field of information retrieval use several metrics to measure the accuracy and efficiency of a pipeline or model. Given a dataset that has labelled relevant query results, for each query we can label a 0 if the model generates no relevant results in the top K and 1 if the model generates at least 1 relevant result. With this $0, 1$ label, we can generate Precision@K and Recall@K, which are the typical classfication metrics across the query set.

The next generation of metrics attempt to improve upon the classification metrics by penalizing how low rank the relevant answers are. Imagine a retrieval solution that had high Precision@10 and Recall@10, but consistently returns the relevant solution below rank 5. This means that the user would have to read many results before deciding which one to accept. It would be beneficial if the relevant result was higher.

An improvement on Precision@K is Mean Average Precision@K (MAP@K). Here is a short

formula set to generate it:

$$AP@K = \frac{1}{r}\sum_{k=1}^{K} Precision@K \cdot rel(k) \quad (1)$$

$$rel(k) = \begin{cases} 0, \text{ if item in position } k \text{ is relevant} \\ 1, \text{ otherwise} \end{cases} \quad (2)$$

$$MAP@K = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{r}\sum_{k=1}^{k} Precision@K \cdot rel(k) \quad (3)$$

where r is the total number of relevant results in the top K recommendation and M is the number of queries in the dataset. As you can see, the precision is weighted by how high in the top K results the relevant result is ranked.

Finally, another metric in the information retrieval space is normalized Discounted Gain Cumulative. A formula for generating is:

$$DGG@K = \sum_{k=1}^{K} \frac{r(k)}{log_2(k+1)} \quad (4)$$

$$nDGC@K = \frac{DGG@K}{IDGC@K} \quad (5)$$

Where the $IDGC@K$ is the function for the ideal ranking based on relevance. This tells a model developer exactly how far from the ideal the ranking is.

3.3 Model Setup and Implementation Details

We benchmarked on three models. One leading choice was made from the three primary families of information retrieval models: Bag of Words, Word Embeddings and Sentence Transformers.

3.3.1 BM25

The BM25 model is a popular Bag of Words ranking function used in information retrieval systems. It calculates the relevance score of a document with respect to a query by considering both the term frequency and the inverse document frequency of the query terms in the document. Specifically, the BM25 model uses a set of tuning parameters to balance the importance of term frequency and inverse document frequency in the relevance score calculation.

We used the industry implementation within Elasticsearch for our testing. This choice was made as ES is a common text database used by businesses in the industry today for their retrieval and NLP use cases.

3.3.2 Dual Embedding Space Model (DESM)

The train dataset was pre-processed into tokens containing words in the lowercase with all the punctuation removed. These tokens were used to train a word2Vec model. Additionally, each article in the json file was saved in a form of a separate document and saved for later during the retrieval tasks. The centroid of all the normalized vectors for the words in the document serving as a single embedding for the whole document was computed and stored in a json file. In this formulation of the DESM, the document embeddings are pre-computed, and at the time of ranking, we only need to sum the score contributions across the query terms. During query time, the word embeddings of the terms in the query are calculated and the average of the sum of the dot product of the query word embedding and the document centroid is calculated and scored according to the cosine similarities.

3.3.3 coCondenser

The coCondenser model is a series of improvements on top of the sentence transformer BERT. First, the Condenser model adds a pre-training architecture on BERT which learns to condense information into the dense vector through language modelling. The coCondenser model then adds an unsupervised corpus-level contrastive loss to warm up the passage embedding space.

Currently, the coCondenser model is the leading information retrieval solution on the MSMARCO benchmark dataset of 100,000 Bing searches with labelled relevance results. Bing searches, however, are a very different corpus when compared to a specific domain like medical data, and the result section illustrates that a model which leads on a general dataset does not necessarily perform well on every dataset.

3.4 Results

Table 1 displays results across datasets and models, with 4 metrics that measure the quality of the top 10 results. The rows in the table are highlighted in green for the best solution on both datasets. DESM In-Out is the best model for both datasets, with coCondenser underperfoming BM25 on NFCorpus and outperforming BM25 on TREC-COVID.

Dataset	Model	MAP@10	nDGC@10	Precision@10	Recall@10
NFCorpus	coCondenser	0.07113	0.22291	0.16656	0.10277
NFCorpus	Elasticsearch BM25	0.12969	0.34281	0.24708	0.16603
NFCorpus	DESM In-In	0.13076	0.34452	0.24904	0.16754
NFCorpus	DESM In-Out	0.14143	0.35387	0.25392	0.17121
TREC-COVID	coCondenser	0.01775	0.72653	0.762	0.01956
TREC-COVID	Elasticsearch BM25	0.01698	0.68803	0.734	0.01907
TREC-COVID	DESM In-In	0.01895	0.74234	0.782	0.02012
TREC-COVID	DESM In-Out	0.01932	0.74932	0.796	0.02185

Table 1: Information Retrieval Results across both datasets.
Amongst the 4 metrics, the net Discounted Gain Cumulative (nDGC) metric is the one that replicates the user experience

3.5 Analysis

Why would a model which leads on a large dataset underperform simplistic solutions on smaller, more domain specific datasets? Put generally, underperformance in information retrieval can be traced down to the data being out-of-domain for the trained model.

Our analysis shows that one source of unfamiliarity to the coCondenser model is the length of the queries. Table 2 illustrates the nDGC metric by model and by query length in tokens in the NFCorpus dataset. As the table indicates, when the coCondenser model is presented with queries that are less than 4 tokens long, it underperforms the other models; when queries are longer, the coCondenser model is much closer in performance to BM25 and DESM. This result is easy to understand, since coCondenser is a sentence transformer model and thus underperforms when presented with one and two word queries. With these single word or short phrase queries, the relevant result is more likely to be the document that directly contains that word, and the power of context that transformers exploit for their performance is irrelevant. However, when the model is applied to phrases that are closer to sentences, coCondenser closes the performance gap and scores an nDGC close to the other models.

But why does it still fall short of the mark even on these longer queries? We come to the second form of unfamiliarity for the coCondenser model: it is untrained on these datasets. Medical dataset's have specific vocabulary that is not common in basic search engine results and cannot necessarily be interpreted by the context of the query without additional training. To interpret the clarity of the responses coCondenser was generating, we averaged

Model	Length	Value
coCondenser	<=3 tokens	0.09045
coCondenser	>3 tokens	0.33586
BM25	<=3 tokens	0.33569
BM25	>3 tokens	0.34967
DESM In-Out	<=3 tokens	0.34967
DESM In-Out	>3 tokens	0.34967

Table 2: nDGC across models and query length for the NFCorpus datasets

the normalized cosine similarity of the top 5 results for each query and compared these numbers versus the same numbers for DESM. Figure 1 illustrates these results. Clearly, coCondenser is very unsure about the relative accuracy from the top rank to the fifth rank. DESM, on the other hand, cuts the cosine similarity of the result nearly in half from the top position to the fifth position. The relative confidence between DESM, which has embeddings specific to this corpus, and coCondenser, which is attempting to extrapolate outside of its training set, illustrates how little the transformer is capable of doing without training on this domain.

4 Related Work

The basic objective of modern Information Retrieval (IR) is to provide the most relevant information to the end user in their query. This task can be divided into two parts: 1) retrieval; and 2) ranking. The first stage retrieves a set of initial documents that are likely to be relevant to the query, and the second stage re-ranks the rank documents based on their relevance score. Many previous publications have covered this problem. Some significant works include the the term weighting methods introduced by Salton and Buckley (1988),

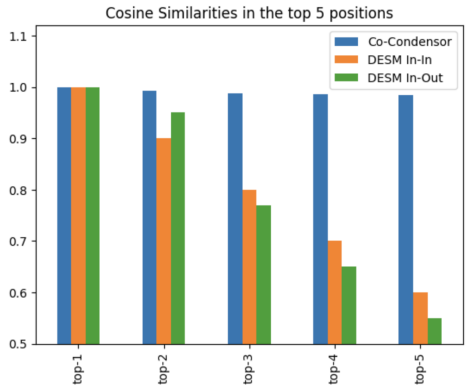

Figure 2: Average top-5 query result cosine similarity across models in the NFCorpus dataset, normalizing to the first result cosine similarity.

Robertson and Jones (1976) in their papers refer to the methods used to assign numerical weights to terms in a document, which are then used to calculate the relevance of a document to a given query. These approaches aim to capture the importance of terms in a document and their discriminative power in distinguishing between relevant and irrelevant documents, and they include approaches such as TF-IDF, BM25(Robertson et al., 1995), and Okapi. Another method of high relevance is the vector space model introduced by Salton (1991) which represents text documents as vectors in a high-dimensional space, where each dimension represents a term in the vocabulary. The similarity between a query and a document is then computed as the cosine similarity between their respective vectors. On these lines, the Dual Space Word Embedding Model was introduced by Mitra et al. (2016) which trains a single layer neural network and uses both the input and output weight projections; we leverage this model in our work. A statistical technique developed by Deerwester et al. (1990) called Latent Semantic Indexing analyzes the relationships between terms and documents to identify patterns of word usage, and it improves retrieval accuracy compared to traditional methods that rely solely on exact keyword matches. Following this work another statistical model called the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) used for topic modeling, which discovers hidden topics in a collection of documents and assigns each document a probability distribution over those topics. It assumes that each document is a mixture of topics, and each topic is a distribution over words. More recent techniques used in document retrieval are pre-trained sentence transformer models like BERT (Devlin et al., 2019) which have gained popularity in recent times due to its ability to encode contextual information of words in a document. Sentence transformer models have achieved state-of-the-art results in various information retrieval tasks, such as question answering, semantic search, and passage retrieval.

5 Conclusion

Optimal performance on information retrieval tasks for small, domain specific datasets appears to require at least some finetuning, even for benchmark leading transformer models. Specifically, leading transformer models tend to under perform more simplistic models on shorter queries that are out of distribution, where its sentence context power cannot be leveraged. Using the cosine similarity view, we can confirm that the coCondenser model is very unsure about which answer is relevant and correct when compared to a sentence embedding model.

What does this mean for those looking to upgrade their business specific document search engines? If your dataset is significantly different than a standard search engine corpus (like random Bing queries), plug and play with transformers is unlikely to produce better results than a BoW or embedding based solution. Training your transformer is likely required to produce better results.

6 Future Work

This work did not attempt to fine tune the transformer to confirm that it can progress beyond a standard search engine solutions. If given more time, this was the next goal we would have prioritized.

In addition, a consideration for information retrieval is that out-of-vocabulary terminology should be handled correclty, which is an advantage that transformers give for free when the query has enough context to answer the question. Traditional models and the DESM model struggle in this area, and handling it would complete the mastery that the DESM model maintains over untuned modern transformers.

Acknowledgements

We would like to thank Professor He He and Nitish Joshi for their inputs throughout the project.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jinhyuk Lee, Wonjin Yoon, Sangwoo Kim, Donghyeon Kim, Sunkyu Kim, Chanyoung So, and Jaewoo Kang. 2019. Nfcorpus: A full-text learning to rank dataset for medical information retrieval. *arXiv preprint arXiv:1910.11445*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2016. Dual embedding space model for document ranking. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 993–996, New York, NY, USA. ACM.

Stephen E Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146.

Stephen E Robertson, Steve Walker, Steve Jones, Michel M Hancock-Beaulieu, M Gatford, et al. 1995. Okapi at trec-3. *NIST Special Publication SP*, 109:109.

Gerard Salton. 1991. Developments in automatic text retrieval. *Science*, 253(5023):974–980.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Lianhui Wang, Dina Demner-Fushman, Ben Carterette, James Allan, Xiaozhong Liu, Dario Taraborelli, Kyle Lo, Laura-Maria Muresan, Lucy Lu Wang, Arman Cohan, Peter R. Elkin, Blair-Goldensohn, Hamed Amjadi, Hengyi Fu, Yanshan Wang, J. Robert B. Lowe, Guido Zuccon, Ferhan Ture, Jaehoon Lee, Shuai Liao, Jingyuan Zhang, Matthew Lease, William Hersh, and Ian Soboroff. 2020. Trec-covid: A special track on covid-19 information retrieval.