Introduction to Information Retrieval - Project 3

Nicholas Drummey
University of Southern Maine
Biddeford, ME, USA
nicholas.drummey@maine.edu

Ryan Reed
University of Southern Maine
Parsonsfield, ME, USA
ryan.reed@maine.edu

ABSTRACT

Stack Exchange is a forum which fosters the discussion of a multitude of field-specific methodologies, tools, techniques, resources, and general Q/A, divided into sub-websites. Within this project, we utilized a number of information retrieval models over the Blender Stack Exchange, as a corpus of user posts. Information retrieval models used included inverted index, boolean retrieval, BM25, LGD, TF-IDF, RM3-BM25, BM25/TF-IDF fusion, and the sentence-transformer model, paraphrase-MiniLM-L3-v2. This study resulted in the conclusion of a trained paraphrase-MiniLM-L3-v2 being the most effective, as a neural network that is trained over the corpus. Another finding in terms of cost-efficiency, is that boolean retrieval was ultimately the most time-cost effective, when including index/train time, all non-neural network models are significantly more time efficient.

The project can be accessed at: Project GitHub Link

CCS CONCEPTS

• **Information systems** → **Presentation of retrieval results**; **Top-k retrieval in databases**; *Combination, fusion and federated search*; **Retrieval effectiveness**; **Evaluation of retrieval results**; **Retrieval efficiency**.

KEYWORDS

MiniLM, Sentence-Tranformer, BM25, TF-IDF, LGD, Boolean Retrieval, Inverted Index, query expansion, IR system evaluation, IR system comparison

1 INTRODUCTION

The sub site of Stack Exchange chosen for this project was the Blender Stack Exchange. In this sub-forum, the specialization of discussion is on the 3D modeling software called Blender. The choice of this sub-forum as a data collection was based on mutual interest by the team, as both team members are familiar with the software, and the implementation of models generated by Blender into game development engines. An additional combined hope of the team is that through our effort on this project, we will gain further practical knowledge in both the realms of information retrieval and 3D modeling. The Blender Stack Exchange can be accessed at https://blender.stackexchange.com/.

This project is built upon two previous projects, in the first, our team implemented Boolean and Inverted Index information retrieval models [5]. In the later project, we constructed more complex IR systems in the form of TF-IDF, BM25, and LGD models, with implementations of BM25-RM3 and TF-IDF/BM25 fusion [6]. In this project, we implement a pretrained sentence-transformer model, namely paraphrase/MiniLM-L3-v2, and perform training over it, collecting results for both the pretrained and fine-tuned runs, and performing evaluation.

Size statistics for the Blender Stack Exchange included 185,140 total documents, 83,738 terms, and 10,959,038 total tokens across all documents. The corpus over the Stack Exchange was composed of user posts, both posted questions and answers.

2 MODEL DESCRIPTIONS

Within this section are descriptions of the models used for the Blender Stack Exchange, including the query expansion model.

2.1 Boolean Retrieval

The Boolean Retrieval implementation takes an operation, AND or OR, to perform and a given query. Breaking the query into specific terms, the model computes document scoring upon comparing the terms present in each document and the terms of the query. The model checks that the index term is either present, or absent, within the document, and increments the score as a result. After computing the list of scoring for each document, for each term, these lists are intersected, forming the results ordered by total score.

2.2 Simple Inverted Index

The simple inverted index implementation performs search with a term-at-a-time approach. The inverted index model first prepares a token-level inverted index, or a dictionary in form (Token : DocumentIndex). With each term of a given query string, the inverted index model increments the scoring for all documents with term occurrences.

2.3 TF-IDF

The TF-IDF, or Term Frequency-Inverse Document Frequency, model uses the statistical measure of which is eponymous with the model, to score the documents within a collection based on a set of query terms. The term frequency of a word in a document is the raw count of the word within the document, and the inverse document frequency is the measure of how frequent a document containing the word appears in the collection. The Formula for the TF-IDF model:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{1}$$

Where t is the term, d is the document, and D is the document collection. The formula for term frequency:

$$tf(t, d) = log(1 + freq(t, d)) \tag{2}$$

The formula for inverse document frequency:

$$idf(t, D) = log(\frac{N}{count(d \epsilon D : t \epsilon d}) \tag{3}$$

A higher TF-IDF score implies that the document is more relevant to the query. [7]

2.4 BM25

BM25, or Best Match 25, is a ranking function that ranks documents according to their relevance to a given query. In many ways, it is an improved version of the TF-IDF model, as it uses term frequency and inverse document frequency. Beyond term frequency and inverse document frequency, it incorporates document length & two tuning parameters, one for controlling scaling of document term frequency and one for controlling the scaling by document length. The formula for Okapi BM25, the specific implementation, is as follows:

$$RSV_d = \sum_{t \epsilon q} \frac{tf_d(1+k)}{k((1-b)+b \cdot (L_d/L_{ave})) + tf_d} \tag{4}$$

Where tf_d equals the term frequency in document d, L_d and L_{ave} refer to the length of document d and the average document length in the collection respectively. The variables k and b refer to tuning parameters for controlling the scaling of the document term frequency and the scaling by document length respectively.

Similar to the TF-IDF model, higher scores indicate higher relevancy with the query. [8]

2.5 LGD

The LGD, or log-logistic distribution, model is a more advanced model that is a combination of two instances of informative content, which are derived from probability distributions. The more the distribution of a word within a document deviates from the average distribution of that word in a collection, the more likely the word is significant for the associated document. This pretext forms the basis of information based information retrieval models, such as LGD.[4] To begin explaining this mathematically, consider the following formula:

$$RSV_{q,d} = \sum_{w \epsilon q \cap d} x_w^q [log((\frac{N_w}{N}) + t_w^d) - log(\frac{N_w}{N})] \tag{5}$$

where the parameters of a word frequency probability distribution are selected as the second term frequency normalization ($t_w{}^d$) and the document frequency. Higher LGD ranking indicates higher relevancy to the associated query.[3]

2.6 RM3

RM3, or relevance model 3, is a technique of query expansion based upon blind feed-back. First, a collection of documents must be ranked by a ranking function, in which a top set of documents become evidence, or feedback, for the relevance model. The relevance model then takes this evidence, and captures the behavior, or pattern, of the returned documents and adapts the associated query based upon the correspondingly computed weights of terms in the evidence. Often, for competitions, the term weights are normalized to sum to one, for this project, this was not chosen.[1] The features for relevance model 3 is as follows:

$$\sum_{q_i \epsilon Q \cap D} log(idf(q_i)) \tag{6}$$

2.7 MiniLM-L3-v2

MiniLM-L3-v2, a sentence-transformer model or approach, is largely composed on Sentence-BERT, and a brief depiction can be seen in Figure 1. The bi-encoder architecture computes sentence embeddings from the query, and from the data in the corpus. These embeddings, or feature vector representations, are then compared pair-wise to compute cosine-similarity. The model performs Wordpiece tokenization, and performs masking token placement (such as [CLS], or [SEP]). The vector representations for the MiniLM models use size 384. Pooling is done through using the mean technique, and the scoring function is through cosine-similarity. The key features between MiniLM and SBERT, are that MiniLM attempts to capture self-attention distributions and value relation. The steps for tokenization and processing into vector representations can be seen in Figure 2.

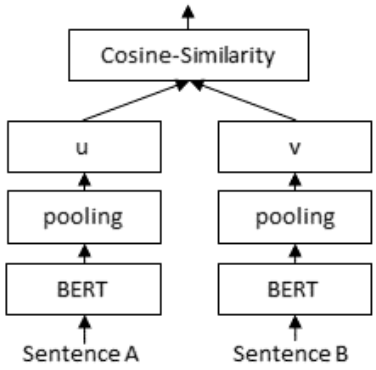

Figure 1: SBERT Bi-encoder architecture.

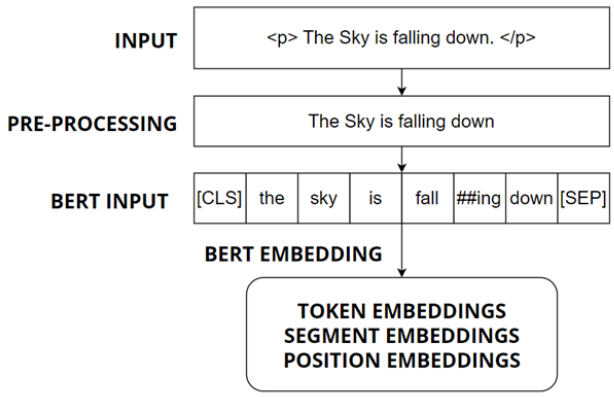

Figure 2: MiniLM-L3-v2 Wordpiece-based tokenization summary.

Self-attention distributions and value relations are found between the probability scores of the elements in 'self' vector representation, the distribution is merely the range of the self-attention values, and value relations are the scores themselves. The general purpose is to create a meaningful representation of the tokenized and masked sequence before it is transformed to an encoding [2]. A representation of these probability scores can be seen as a matrix in figure 3.

Table 1: IR Model Search Effectiveness Results

Model	P@5	nDCG@5
Boolean Retrieval	0.350	0.707
Inverted-Index	0.310	0.835
TF-IDF	0.450	0.858
BM25	0.430	0.851
LGD	0.490	0.871
TF-IDF/BM25 Fusion	0.450	0.875
RM3-BM25	0.390	0.883
Pretrained MiniLM	0.480	0.866
Fine-Tuned MiniLM	0.530	0.922

Table 2: IR Model Time Efficiency Results

Model	Index/Train Time	Search Time
Boolean Retrieval	1.3 min	20 ms
Inverted-Index	2.1 min	35.6 sec
TF-IDF	N/A	2.8 min
BM25	N/A	2.4 min
LGD	N/A	3.2 min
TF-IDF/BM25 Fusion	N/A	1.5 min
RM3-BM25	N/A	3.4 min
Pretrained MiniLM	N/A	2.25 min
Fine-Tuned MiniLM (20 epoch)	4.8 min	2.25 min

An important to note with the projects implementation of the MiniLM-L3-v2 model, is the parameters. Our team trained the model over 20 epochs, taking around 1 hour and 39 minutes to train. The training batch size was 64 sentence pairs, while the margins were 0.5. We chose a maximum sequential length of 128, to avoid quadratical scaling which can drastically affect performance.

	Hello	I	love	you
Hello	0.8	0.1	0.05	0.05
I	0.1	0.6	0.2	0.1
love	0.05	0.2	0.65	0.1
you	0.2	0.1	0.1	0.6

Figure 3: Matrix representation of a sequence's probability scores (Self-attention values, an example).

Table 3: Boolean Retrieval - Top 3 Search Results

Post ID	Relevance	Link
55991	0	link
69306	2	link
146393	2	link

Table 4: Inverted Index - Top 3 Search Results

Post ID	Relevance	Link
69306	2	link
55991	0	link
146393	2	link

Table 5: TF-IDF - Top 3 Search Results

Post ID	Relevance	Link
146393	2	link
69658	2	link
80137	0	link

Table 6: BM25 - Top 3 Search Results

Post ID	Relevance	Link
146393	2	link
69658	2	link
80137	0	link

3 EVALUATION

Within this section is the discussion of search results, and the comparison of results between different models.

3.1 Search Results

Table 1 features the normalized discounted cumulative gain and precision, at cut 5 results of each models run. While Table 2 features the time cost of training/indexing and the time cost of search. All models proved time-cost efficient, and varied in effectiveness, but all models performed marginally well for the scope of this project. We also provide the cut 3 posts returned, for each model for query 1 as examples. Query 1 is "Blender 2.8 - weathering effects", and can be accessed at: Link

We observe from tables 3-8, that most of the information retrieval models returned similar results for this query, with the major differences being differences in rank based upon different score computations. To contrast this similarity, we point to the untrained and trained, or pretrained and fine-tuned, MiniLM-L3-v2 models. The results returned by these models, as shown in tables **??** and **??**, not only stray from the similar results of the other models, but have comparable results, an important notion is that the posts retrieved are at a minimum, partially relevant, whereas the others return non-relevant posts. In the following section, we delve deeper

Table 7: LGD - Top 3 Search Results

Post ID	Relevance	Link
146393	2	link
69658	2	link
80137	0	link

Table 8: RM3-BM25 - Top 3 Search Results

Post ID	Relevance	Link
55991	0	link
69306	2	link
146393	2	link

Table 9: Pretrained MiniLM - Top 3 Search Results

Post ID	Relevance	Link
213831	1	link
168903	1	link
194435	1	link

Table 10: Fine-tuned MiniLM - Top 3 Search Results

Post ID	Relevance	Link
68099	1	link
203550	1	link
129764	2	link

into evaluation, namely as it pertains to comparisons between the information retrieval models.

3.2 Model Comparisons

In Figure 4, we display a visualization of the precision at cut 5 to give a clearer depiction of the differences between models. We can see a general range in the interval (0.310, 0.530), with most models falling around 0.45. The outlying models in regards to precision at cut 5 are boolean retrieval, inverted index, and the trained MiniLM models at 0.350, 0.310, and 0.530, respectively. It's clear within this context that the fine-tuned MiniLM model is the most effective, while boolean retrieval and inverted index both performed poorly in regards to the grouping of models. Interestingly, the log-logistic distribution performed slightly better than the pretrained MiniLM model.

Figure 5 shows the pattern of normalized discounted cumulative gain at cut 5, and the results are significantly closer than for precision at the same cut, with a grouping around 0.85. The only distinct outliers are the boolean retrieval and the trained MiniLM models, at 0.707 and 0.922. This, compared with the model results for precision, points woards boolean retrieval being significantly less accurate when compared to other models, and provides support for the greater effectiveness of a trained MiniLM model.

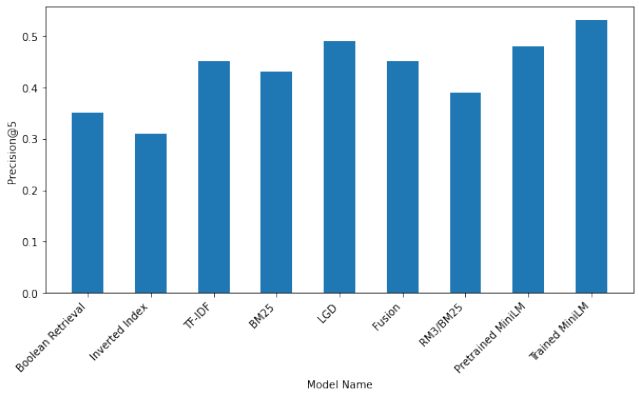

Figure 4: Visualization of Precision@5 for all Models.

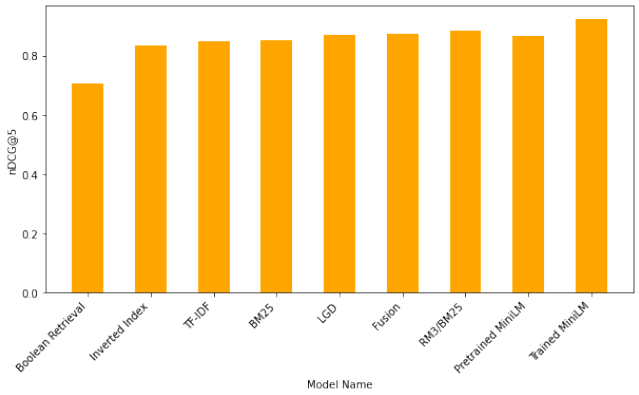

Figure 5: Visualization of nDCG@5 for all Models.

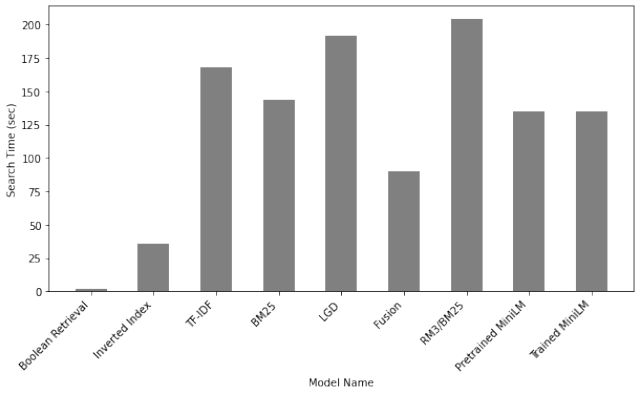

Figure 6: Search Time in Seconds, for all Models.

An interesting, and important insight into the models utilized within this project, is the time-cost efficiency. This is also the least uniform and the most differing set of recorded measurement for the group of models. While previously, we've discussed the failure of effectiveness for boolean retrieval, the time efficiency for this model, however, is not nearly paralleled by any other model. Other

models with positively notable time efficiency for a 20 query search include the TF-IDF/BM25 fusion and the inverted index models. Negatively notable time efficient models include TF-IDF, LGD, and the RM3-BM25 models. Previously notable for high effectiveness, the pretrained and trained MiniLM models run just over an average time efficiency, signifying an effective time efficiency trade-off for remarkable effectiveness.

4 CONCLUSION

In conclusion, after familiarizing ourselves with the nine unique models and analyzing the runs over the collection, there is one model that performed better than the others overall, by a somewhat large margin. The trained MiniLM-L3-v2 model not only had the best effectiveness, but an only slightly above average time efficiency. This model's results came at the cost of training time, which in total took around 1 hour and 39 minutes, but once the model was trained and saved, remained the objectively best model. However, despite this, there is a reasonable amount of support for the other models being useful in seperate situations. Using boolean retrieval as an example, while the model is extremely simple and tends to have poorer accuracy, it is highly time and resource efficient.

All members of the team found this project to be highly beneficial to our understanding of IR models, IR analysis, and IR model comparing, as well as the journey into more advanced models than our previous projects. Throughout previous projects we've put in practice with important tools, such as PyTerrier and the trec_eval software. An important discipline gained by teammates during this project, was the exploration into scientific papers, primarily on LGD and RM3, as only the source papers provided information about these subjects. Other notable skills that we feel we've grown in over the course of this project include report writing with overleaf, presentation, and Blender as we worked over the collection further.

In regards to project 3, where we are working with a neural network model and a push into far more advanced topics, our team can agree on our experiences. Adding to our repertoire of information retrieval topics, are not only working with neural network systems, but using hugging face and studying not only documentation, but scholarly articles and scientific journals, to try and gain a much more reinforced understanding of the more advanced techniques that we've had to work with. Information retrieval topics are arguably hard to digest, but exhaustive searching and putting forth effort to understand on a more fundamental level, are important skills that we've especially honed in this section of our overall semester-long project.

REFERENCES

[1] Nasreen Abdul-Jaleel and James Allan. 2004. UMass at TREC 2004: Novelty and HARD. (March 2004). https://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf

[2] Nikolas Adaloglou. 2020. *How Attention works in Deep Learning: understanding the attention mechanism in sequence models.* Retrieved December 12, 2022 from https://theaisummer.com/attention/#self-attention-the-key-component-of-the-transformer-architecture

[3] Stéphane Clinchant and Eric Gaussier. 2010. Information-Based Models for Ad Hoc IR. (Jan. 2010). https://doi.org/10.1145/1835449.1835490

[4] Stéphane Clinchant and Eric Gaussier. 2011. A Log-logistic Model for IR. *Springer Verlag* 14, 1 (2011), 5–25. https://doi.org/10.1007/s10791-010-9143-7

[5] Nicholas Drummey. 2022. *Project Part 1.* Retrieved December 12, 2022 from https://github.com/CousinNic/cos-470-data-retrieval/tree/main/Project1

[6] Nicholas Drummey and Ryan Reed. 2022. *Information Retrieval Project 2.* Retrieved December 12, 2022 from https://github.com/CousinNic/COS470_Information-Retrieval_Project2

[7] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2009. *An Introduction to Information Retrieval.* Cambridge University Press, Cambridge, England.

[8] Wikipedia. 2022. *Okapi BM25.* Retrieved November 10, 2022 from https://en.wikipedia.org/wiki/Okapi_BM25#The_ranking_function