50.045 Information Retrieval

Project Report

12 August 2021

Xie Han Keong (1003876, hankeong_xie@mymail.sutd.edu.sg)
Jisa Mariam Zachariah (1003638, jisa_zachariah@mymail.sutd.edu.sg)
Lucas Ng Yi Leang (1003478, lucas_ng@mymail.sutd.edu.sg)

Abstract

This report details our attempts at building a search engine for retrieving news articles. The aim of our project is to find the most relevant news article(s) in a collection given a search query. We explored five different approaches and evaluated their performance based on mean reciprocal rank and latency. We found that SBERT had the highest mean reciprocal rank followed by BM25, while QLM had the lowest latency followed by BM25. We conclude that SBERT is the most effective, QLM is the fastest, and BM25 is the most well-rounded model for news retrieval.

Introduction

Information retrieval is a critical task in today's digital age due to the overabundance of information. In a world with endless news sources and articles, the role of a search engine is indispensable—namely, when given a search query, its task is to return articles that are most relevant to that query. However, ranking news articles is not an easy task, and many search engines occasionally fail to return good results. To find out why, we embarked on this project to gain a first-hand understanding of what it takes to build a great search engine like Google.

Dataset

Raw Dataset

We obtained a raw dataset from a company that we were partnered with under the Capstone programme to develop a news monitoring platform. The dataset comprised 285,770 news articles which were scraped from various news sources (See Figure 1). Of all the given features, we only used ID, title, and content (See Table 1). Furthermore, we narrowed the dataset to 1,000 randomly selected documents so that we could prototype quickly.

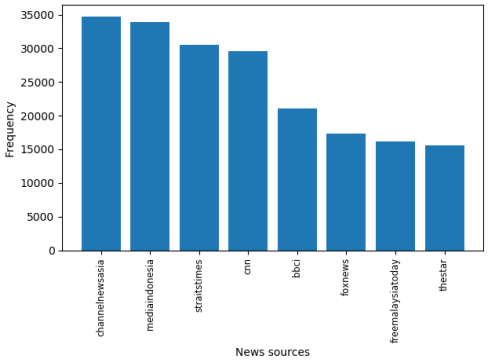

Figure 1: Distribution of frequencies of the top 8 news sources

Table 1: Features of the raw dataset (only relevant features shown)

Feature	Type	Sample
ID	integer	3820309
Title	string	What we know about the 6 new cases of novel coronavirus in Singapore
Content	string	Six patients in Singapore, including four with no recent travel history to China, have tested positive for the novel coronavirus, the Ministry of Health (MOH) announced on Tuesday (Feb 4). The four cases of local transmission are linked to travellers from China, the ministry said…

Evaluation Dataset

As the raw dataset did not have any relevance labels, we hand-crafted an evaluation dataset using 50 of the randomly selected documents. For each document, we manually wrote a query that is relevant to it based on our individual judgement, and added the query along with the document ID and title to the evaluation dataset (See Table 2). One caveat is that we assumed for each query that the respective document is the only relevant one in the whole collection. Of course, other documents might be relevant too. However, we still used this naive approach due to lack of time and resources.

Table 2: Features of the evaluation dataset

Feature	Type	Sample
ID	integer	1426072
Title	string	12 Palestians killed in clashes at Israel-Gaza border, Middle East News & Top Stories
Query	string	israel gaza border clashes

Approaches

In total, we explored five approaches for news retrieval (See Figure 2). In the following sections, we detail the concepts and process flow behind each approach.

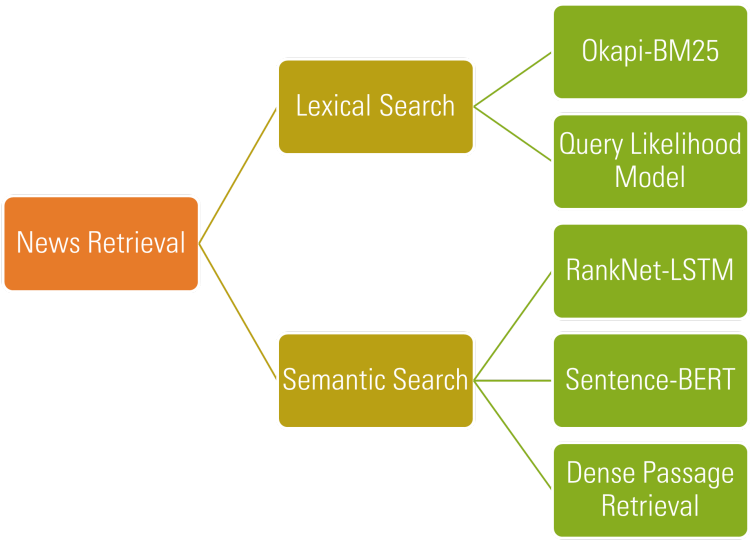

Figure 2: News retrieval approaches that we explored

Okapi BM25

Okapi BM25 is one of the most widely used ranking algorithms today (Feng et. al., 2021). It is a probabilistic bag-of-words model that improves on TF-IDF by paying more attention to the term frequency and document length (Manning, Raghavan & Schütze, 2008). Specifically, it imposes a diminishing return on term frequency and a score handicap on longer documents (Liu, 2018). We used an inverted index to speed up the process of finding candidate documents. See Figure 3 for the process flow of our BM25 search.

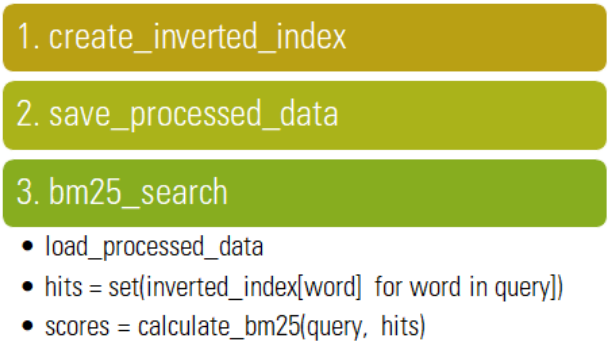

Figure 3: BM25 search process flow

Query Likelihood Model

The query likelihood model (QLM) is a method in which each document is represented by a language model (LM)—most commonly, a unigram language model (Manning, Raghavan & Schütze, 2008). In QLM, documents are ranked by the probability that a query would be observed given the document's language model (Ibid.). This can be estimated by taking the product of the document term frequency divided by document length over all terms in the query (Ibid.). We used add-1 smoothing to avoid zero probabilities and linear interpolation to bring in an IDF-like characteristic. See Figure 4 for the process flow of our QLM search.

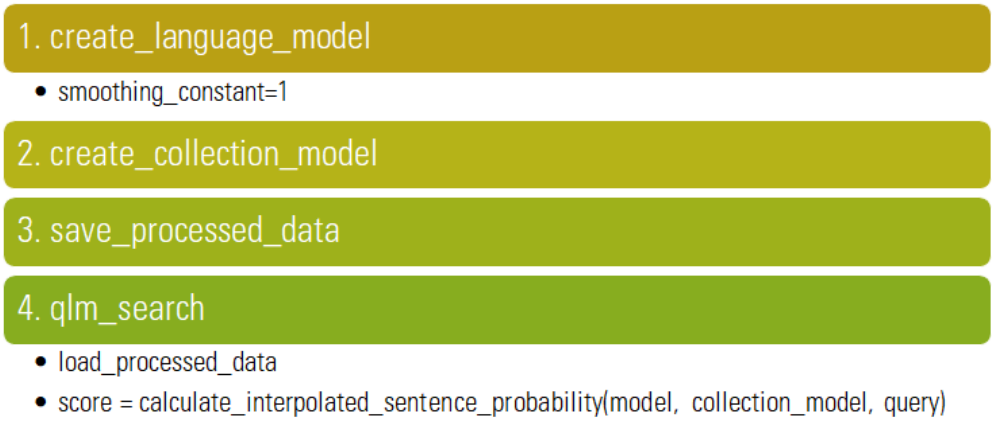

Figure 4: QLM search process flow

RankNet-LSTM

RankNet-LSTM is based on RankNet, which is a Learning To Rank (LTR) model that uses a pairwise approach to train a neural network to rank documents. During inference, the query and document are embedded using GLoVe and passed into their respective encoders. Their encoded representations are then concatenated and passed through a feedforward layer to generate a relevance score (See Figure

5). During training, the score for a relevant document and an irrelevant document are computed and the difference is passed into the RankNet loss function (Burges et. al., 2005) to compute the loss.

For RankNet-LSTM, we experimented with using LSTMs as the encoders instead of multi-layer perceptrons in the classic RankNet. Also, we trained two separate LSTMs for the query and document respectively as they are asymmetric (a query tends to be much shorter than an article). Since we did not have labelled data, we generated a pseudo-labelled dataset using BM25 by taking the top 5 documents as positive samples and 5 other random documents as negative samples and assigning the documents with scores above the 80th percentile as relevant. See Figure 6 for the process flow of our RankNet-LSTM search.

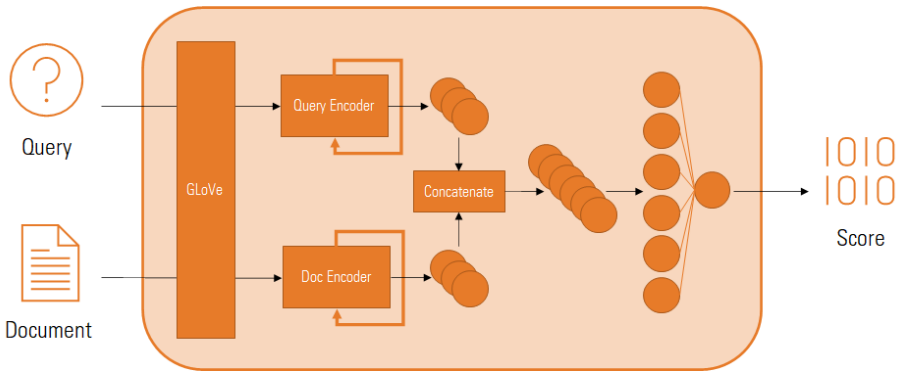

Figure 5: RankNetLSTM inference flow

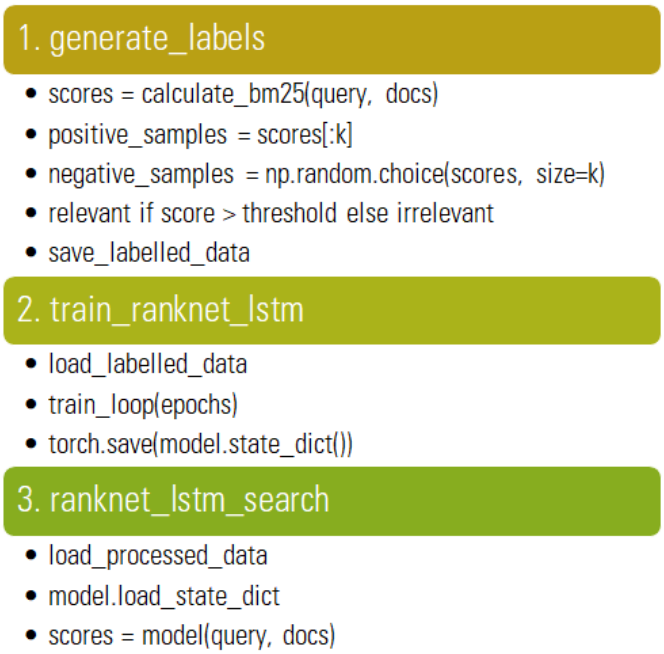

Figure 6: RankNetLSTM search process flow

Sentence-BERT

Sentence-BERT (SBERT) is a modification of BERT using siamese and triplet networks (Reimers & Gurevych, 2019). While BERT can be very effective at predicting sentence similarity, it uses a cross-encoder approach, where two sentences are passed through a transformer to obtain a similarity score (Reimers, 2021). The problem is that the number of combinations required increases quadratically. For instance, if a collection contains 10,000 documents, it would require almost 50

million comparisons to determine the two most similar sentences, which would take about 65 hours for BERT to complete (Reimers & Gurevych, 2019). To rectify this issue, SBERT was developed.

The SBERT architecture is based on a bi-encoder approach, which uses the same BERT model to encode two sentences one after another. The pooler output of BERT is used as the embedding for a sentence (See Figure 7). With these embeddings, a similarity measure such as cosine similarity can be used to obtain a similarity score. The advantage of this approach is that it can be performed in parallel, leading to a much higher throughput as compared to the classical cross-encoder approach used in BERT. Using SBERT, we adopted a retrieve and re-rank approach to rank documents efficiently and effectively. First, we used a SBERT bi-encoder to retrieve a set of relevant documents, taking advantage of its efficiency. Thereafter, we used a BERT cross-encoder to re-evaluate the scores of each document, taking advantage of its effectiveness. Also, we embedded the documents beforehand to reduce runtime overhead. See Figure 8 for the process flow of our SBERT search.

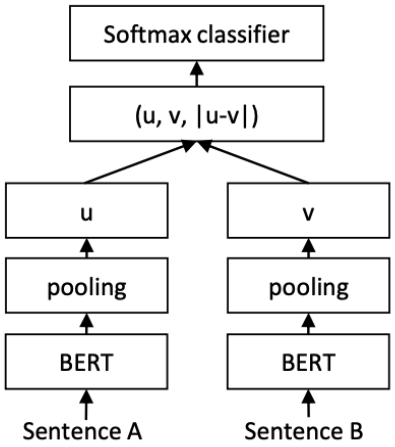

Figure 7: SBERT bi-encoder architecture (Reimers & Gurevych, 2019)

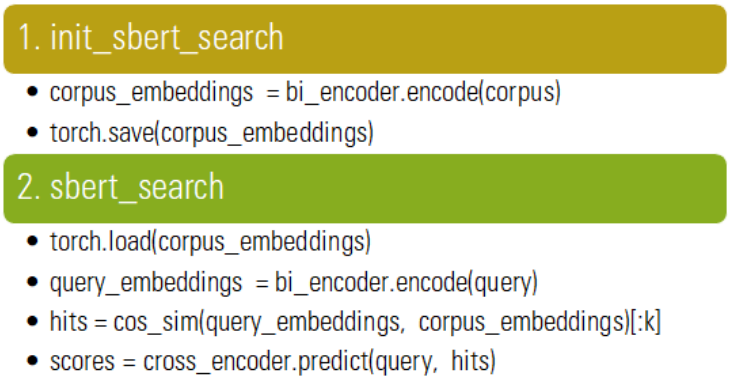

Figure 8: SBERT search process flow

Dense Passage Retrieval

Dense Passage Retrieval (DPR) uses a dual-encoder approach (Karpukhin et al., 2020) to encode a query and document separately—similar to what we did for RankNet-LSTM. However, the difference is that no GLoVe embeddings are used, and instead of using two independent LSTM networks as the query and document encoders, two independent BERT networks are used; and instead of using a concatenation followed by a feedforward layer to calculate the relevance score, dot product is used (See Figure 9). Similar to SBERT search, we embedded the documents beforehand to reduce runtime overhead. See Figure 10 for the process flow of our DPR search.

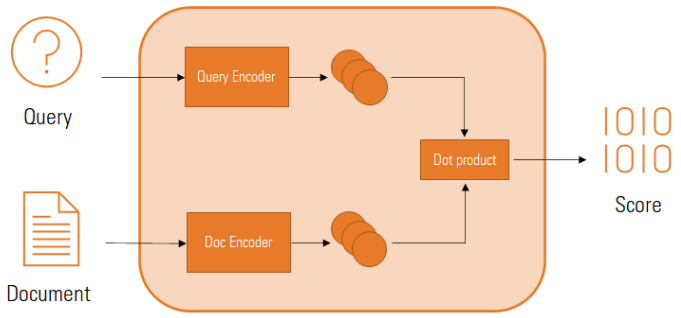

Figure 9: DPR inference flow

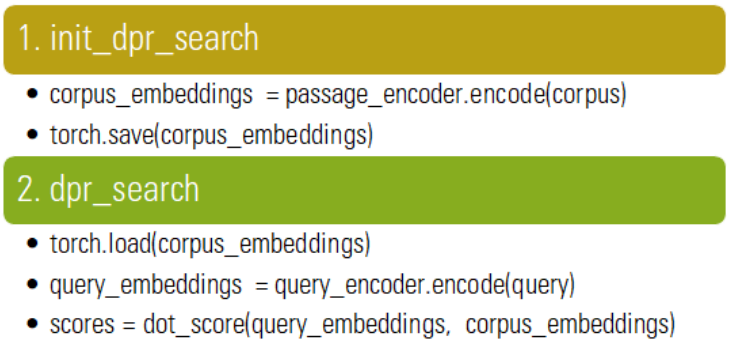

Figure 10: DPR search process flow

Results

We computed the mean reciprocal rank (MRR) and latency (See Appendix A) to evaluate the performance of each model using our evaluation dataset. Note that MRR is equivalent to the mean average precision (MAP) due to the fact there is only one relevant document for each query in our evaluation dataset. We report the results for each model in Table 3 below.

Table 3: Results for each model

Model	**MRR**	**Latency**
BM-25	0.85	25.56
QLM	0.67	**116.08**
RankNet-LSTM	0.00	2.54
DPR	0.56	8.20
SBERT	**0.87**	0.23

Analysis

SBERT had the highest MRR followed by BM25, while QLM had the best latency (as measured in queries per second) followed by BM25. These results are not surprising as semantic search methods such as SBERT have greater ranking potential than lexical search methods like BM25 and QLM. This is because unlike lexical search, semantic search is able to understand the meaning of a sentence, which allows it to pick up on synonyms and match sentences with similar meaning but different words (Reimers, 2021).

Although QLM had the best latency, its MRR was not as good as compared to BM25, which was on par with SBERT. However, although SBERT had the best MRR, it also had the worst latency. This could be in part due to cold start, which is the time taken to load a model onto the GPU for the first time. It could also be due to the high computation cost of the retrieve and re-rank approach, which runs two transformers sequentially (the bi-encoder to encode the query and the cross-encoder to re-rank the results). This is in contrast to DPR, where only one pass of the query encoder is needed. Since BM25 was second in both MRR and latency, we can conclude that SBERT is the most effective, QLM is the fastest, and BM25 is the most well-rounded model for news retrieval.

Of course, these results are all based on the validity of our evaluation dataset which has many flaws. An evaluation dataset that is larger and contains more relevance labels will give a better reflection of the performance of each model.

Challenges Faced

Labelling the data manually

Due to the lack of relevance data, we needed to manually generate queries and the relevance labels for our evaluation dataset. This process was tedious, time-consuming, prone to inconsistencies due to differences in human judgement, and certainly not scalable. It really shows how useful clickthrough data can be for training a LTR model.

Training the RankNet-LSTM model

The RankNet-LSTM model performed badly because it was unable to learn from the small training dataset. This caused it to overfit and return similar scores for each document regardless of the query. This is in contrast to DPR and SBERT, which were pre-trained on large datasets such as SNLI (Reimers & Gurevych, 2019) and Wikipedia (Karpukhin et al., 2020). Clearly, more training data is needed to achieve better results.

Future Works

Scaling up

Our work was based on a small subset of 1,000 documents only. We chose to do this so that we could prototype quickly. Future work could focus on scaling our work up to the original dataset—which contains more than 285,000 documents—and beyond. In reality, news articles are continuously produced every single day and the role of a search engine should not stop at just one dataset.

Hybrid lexical/semantic search

For the retrieve and re-rank approach, we only explored using the SBERT bi-encoder to retrieve and the BERT cross-encoder to re-rank. Future work could focus on using BM25 to retrieve instead to test if it can achieve better or faster performance.

Conclusion

In this project, we explored five different approaches for news retrieval and found that SBERT performed the best. However, we also found that it was very slow as compared to lexical search methods such as BM25 and QLM, which have much better latencies and comparable mean reciprocal

rank. Hence, we conclude that SBERT is the most effective, QLM is the fastest, and BM25 is the most well-rounded model for news retrieval.

Appendix

Appendix A: Metrics

To compute the MRR of each model, we used the following formula, where Q is the total number of queries in the evaluation dataset, and $rank_i$ is the rank of the first relevant document for query i.

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}$$

To compute the latency of each model, we tracked the time taken for each query and took the reciprocal of its average.

Appendix B: Links

Code repository: https://github.com/han-keong/NewsRetrievalEngine

Presentation slides: https://docs.google.com/presentation/d/1hBrQcF0gQFGNP9AfFnEFUYVhxD1QACHrFzdzcXqUeWA/edit?usp=sharing

References

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent | Proceedings of the 22nd international conference on Machine learning. Retrieved 12 August 2021, from https://dl.acm.org/doi/10.1145/1102351.1102363

Feng, Y., Saelid, D., Li, K., Gao, R., & Shah, C. (2021). University of Washington at TREC 2020 Fairness Ranking Track. Retrieved 12 August 2021, from https://arxiv.org/abs/2011.02066

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., & Edunov, S. et al. (2020). Dense Passage Retrieval for Open-Domain Question Answering. Retrieved 12 August 2021, from https://arxiv.org/abs/2004.04906

Liu, E. (2018). bm25_intro. Retrieved 12 August 2021, from http://ethen8181.github.io/machine-learning/search/bm25_intro.html

Manning, C., Raghavan, P., & Schütze, H. (2008). Estimating the query generation probability | Introduction to Information Retrieval. Retrieved 12 August 2021, from https://nlp.stanford.edu/IR-book/html/htmledition/estimating-the-query-generation-probability-1.html

Manning, C., Raghavan, P., & Schütze, H. (2008). Okapi BM25: a non-binary model | Introduction to Information Retrieval. Retrieved 12 August 2021, from https://nlp.stanford.edu/IR-book/html/htmledition/okapi-bm25-a-non-binary-model-1.html

Manning, C., Raghavan, P., & Schütze, H. (2008). Using query likelihood language models in IR | Introduction to Information Retrieval. Retrieved 12 August 2021, from https://nlp.stanford.edu/IR-book/html/htmledition/using-query-likelihood-language-models-in-ir-1.html

Reimers, N. (2021). Retrieve & Re-Rank — Sentence-Transformers documentation. Retrieved 12 August 2021, from https://www.sbert.net/examples/applications/retrieve_rerank/README.html

Reimers, N. (2021). Semantic Search — Sentence-Transformers documentation. Retrieved 12 August 2021, from https://www.sbert.net/examples/applications/semantic-search/README.html

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Retrieved 12 August 2021, from https://arxiv.org/abs/1908.10084