Academic Year 2023/2024

Biomedical Informatics

POLITÉCNICA

Efficient Self-Supervised Metric Information Retrieval: A Bibliography-Based Method Applied to COVID Literature

Paper Review

Gruppo XX

Andreas Brummer

Lorenzo Leuzzi

Michele Simeone

1 Background

2 Methods

The model used is the CORD-19 dataset. The idea is to use a deep metric learning model and use the bibliography of a paper as a vector of real numbers representing a high-dimensional point. This feature of the latent space allows the comparison between them and the definition of hidden relationships, using the spatial distance as a metric for their semantic similarity.

2.1 Training Set Creation

First, they create a sparse matrix M_{DXC} where D are the papers in the dataset and C are the cited papers(that may not be in the dataset). This matrix is sparse and each cell d, c is set either to 0 or 1:
1 if document d contains c in its reference list, 0 otherwise. To compare the documents we should enquire about the existence of First or High order relationships. If we climb the relationships graph we can discover the existence or a relationship between the 2 documents and use the distance as a weight. To make the hidden information emerge they use the singular value decomposition(SVD,$k = 1024$). The resulting latent space **L** is used in combination with the Cosine Distance to study relationships between papers. To create the training set for each of the papers (P_q, $q = 1, ..., D$) three negatives are chosen (P_n, using the bibliography vector).

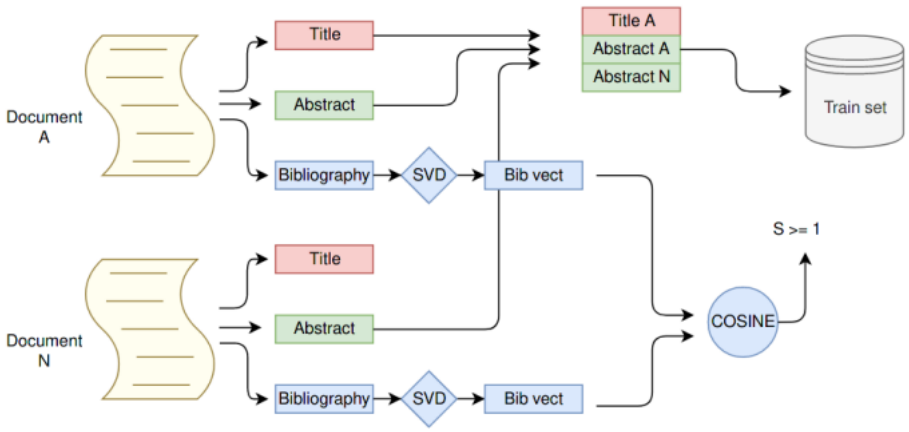

Figure 2.1: Caption

2.2 Loss Function

They use the triplet *loss function* as an objective function for the training.

$$L = max(dp - dn + m, 0) \tag{1}$$

The function takes 3 elements (e_q^t, e_q^a and e_n^a), which are the embeddings of the title and the abstract of the query paper, and the embedding of the abstract from the negative paper. The terms dp and dn are the Euclidean distance between e_q^t, e_q^a and e_q^t, e_n^a respectively. m represents instead the margin which means how close informative negatives have to be.

This is how the Euclidean distance is calculated:

$$d(P, P') = \sqrt{\sum_{i=0}^{i=|P|} (P_i - P_i')^2} \tag{2}$$

The model is trained to generate embeddings and put them closer if they come from the same paper. This provides a link between a title and a long text like an abstract. They used SciBERT to be trained. The text of the Title and the two abstracts is concatenated and then tokenized ($I^S_{ID_S}$). The result $Y^S = M(I^S_{ID_S})$ is a matrix 512x768.

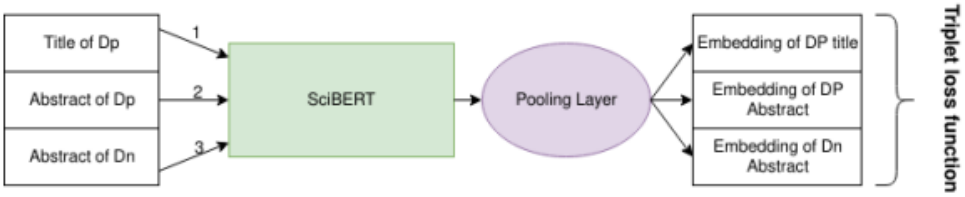

Figure 2.2: Caption

2.3 The IR system

The IR system is called SUBLIMER. The architecture can be divided into 3 modules:

- The **indexer** that uses two data structures for indexing(TF-iDF Vectors and Neural Document Embeddings)
- The **retriever** uses the natural language query to obtain two scores that are combined linearly to obtain a rank of the documents.
- The **reranker** uses the top K document, considering also the full text to sort them a second time.

2.3.1 The Indexer

For all the documents two representations are created. The first one produces a neural embedding using SciBERT fine-tuned on CORD-19 as explained before. It takes in input t_d and a_d (title, abstract of $d \in \mathbf{D}$) and outputs the embedding e_d. For the second, first, they create a bag of words obtained by tokenizing the titles and the abstracts. The iDF is calculated as follows:

$$iDF(w_i) = ln\left(\frac{N - n(w_i) + 0.5}{n(w_i) + 0.5} + 1\right) \tag{3}$$

For each vector is created also a tf_d (term frequency vector).

2.3.2 The Retriever

The query is transformed in the embedding e_q and also the tf-idf vector is constructed (v_q). The embedding of the query is compared to the embedding of the whole documents in $\mathbf{D}$ using the CosSim function to obtain a score s_n.

$$s_n(q, d) = CosSim(e_q, e_d) \tag{4}$$

$$CosSim(a, b) = \frac{\sum_{i=1}^{d} a_i b_i}{\sqrt{\sum_{i=1}^{d} a_i^2}\sqrt{\sum_{i=1}^{d} b_i^2}} \tag{5}$$

Where d is the dimension of the vectors. A second score sb is computed according to the BM25 Okapi.

$$s_b(q, d) = \sum_{i=0}^{|q|} IDF(q_i) \cdot \frac{f(q_i, d) \cdot (k + 1)}{f(q_i, d) + k \cdot (1 - b + b\frac{|d|}{avgdl})} \tag{6}$$

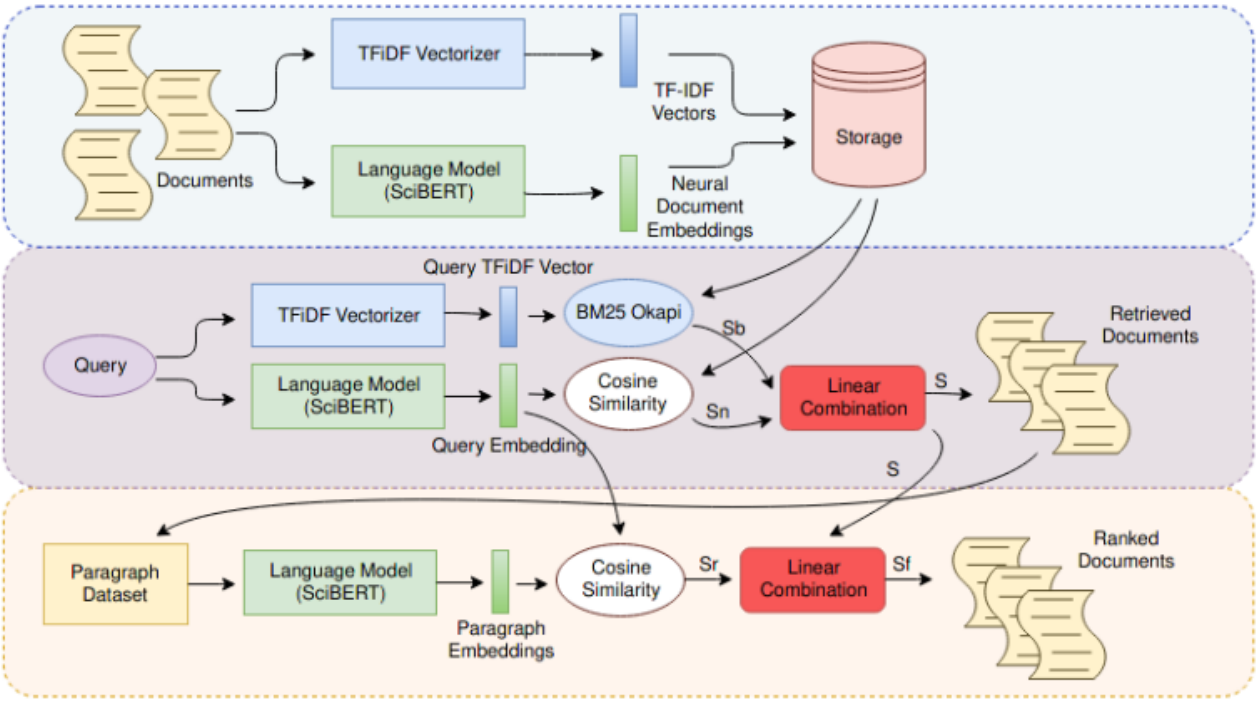

Figure 2.3: 3. The picture shows the entire system from the dataset creation to the ranked documents

Finally the two scores are combined to obtain the final score s

$$s = \alpha \cdot s_n + (1 - \alpha) \cdot s_b \tag{7}$$

2.3.3 The Ranker

For the top p documents the score is recomputed. For each of them, more embeddings $(e_i^p, i = 0, ..., n$ having n paragraphs and 0 representing the abstract)are created using the title and one paragraph or the abstract. After this, the similarity is computed for each embedding and only the maximum score is kept $s_{r_j}, j = 1, ..., p)$. This score is linearly combined with the previous one to obtain the final one.

$$s_f = \beta \cdot s + (1 - \beta) \cdot s_r \tag{8}$$

2.4 Language Model Fine Tuning

Let $y_q =< d_0, ..., d_n >= IR(q)$ be the result of the IR system for a query q. The first 3 results were selected as positive results the last 15 as negative. After pairing each positive with a negative a new training set is created and the old model is fine-tuned with that set.

3 Results

The tests were performed against Co-Search and COVIDEX, state-of-the-art for CORD19 IR. The competitors were trained with different labeled datasets(not only CORD19). For the evaluation, the TREC-COVID Test Set was used.

3.1 Evaluation Metrics

3.1.1 Precision

$$P@N = \frac{|relevant\ documents\ in\ top - N|}{N} \tag{9}$$

3.1.2 nDCG

Normalized discounted cumulative gain performs

$$nDCG@N = \frac{1}{Q}\sum_{q=0}^{Q}\frac{DCG^q}{IDCG^q} \tag{10}$$

where q is the number of queries,

$$DCG^q = rel_1^q + \sum_{i=2}^{N}\frac{rel^q}{log_2(i)} \tag{11}$$

and rel_i^q is the relevance of the retrieved document in position i with respect to the topic q. The $IDCG^q$ is the ideal DCG or the highest possible DCG

3.1.3 MAP

Is the average precision of the document set.

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q}P_q(R)dR \tag{12}$$

3.1.4 Bpref

It checks how frequently irrelevant documents are retrieved before relevant.

$$Bpref = \frac{1}{R}\sum_{r=1}^{R}1 - \frac{|n\ ranked\ higher\ than\ r|}{R} \tag{13}$$

where R is the number of judged relevant documents, r is a relevant retrieved document, n is the number of irrelevant retrieved documents.

3.2 IR Results

The comparison was made first between the model alone and the entire IR system. It was trained with different methods. First, using the triplet loss function but without using bibliography information. (the negatives were selected randomly). In the second case, the bibliography was considered and both triplet loss function and multi-similarity function were used. The second 2 performed better than the first case as expected, confirming that the bibliography brings information in this context. The triplet loss function obtained slightly better performance and was used for the comparison with the state-of-the-art.

The second table shows how the SUBLIMER performed better in 2 metrics than the state-of-the-art. This proves the quality of the solution but the objective in this case was not to overperform the other solutions but to get similar results without using labeled training sets and using fewer parameters. Finally, in Table 3, they present how single components of the information-retrieval system contributes to the final score.

3.3 System Size Comparison

The SUBLIMER has also way fewer parameters compared to the state-of-the-art (at least one order less).

Table 3.1: Caption

3.4 Bibliography Embeddings Evaluation

The quality of the bibliography embeddings were evaluated by using labeled document taken from TREC-COVID. Obviously, good embeddings will place closer papers on the same topics. 1000 random pairs were selected from all the TREC-COVID labeled documents with bibliography information. After, 2500 pairs of papers with relevance two(R2) and 2500 pairs with relevance one (called R1) to the same topic was selected. Then they created the bibliography embedding for each element and computed the cosine distance for each pair. The results proved the supposition was correct.

4 Discussion and conclusion

5 Proposal for improvements and future work