Name: Simple Applications of BERT for Ad Hoc Document Retrieval

Paper Link: https://arxiv.org/pdf/1903.10972.pdf

Github: https://github.com/castorini/birch

A. Introduction.

- Confronts the issue of BERT having to have 512 input tokens only. This is done by applying inference on sentences and then aggregating these scores to find document level score.
- Authors realise that the BERT based system that has seen massive improvements in the QA field can be remodelled to function as a **Document retrieval system**.
- Uses **BERTserini** network that is used for open domain Question Answering.
- Evaluate the approach on TREC Microblog Tracks and TREC 2004 Robust Track
- Authors claim first ever application of BERT to ad-hoc document retrieval.

B. Description

a. The paper **DOES NOT focus on fine-tuning** the BERT architecture for Adhoc document retrieval given only a document and its document level ground truth. Rather it **focuses on how to combine sentence level scores into a document score.**

b. **2 Steps**:

i. **Anserini IR toolkit**: provides top 1000 documents based on Query Likelihood and RMS relevance feedback (document can consist of multiple sentences)

ii. The text of the retrieved document is fed in **BERT-base** classifier and the BERT scores are combined via **LINEAR INTERPOLATION.** This is done 1 document at a time.[d_1, d_2 …. D_1000 returned by step 1]

c. **Architecture**

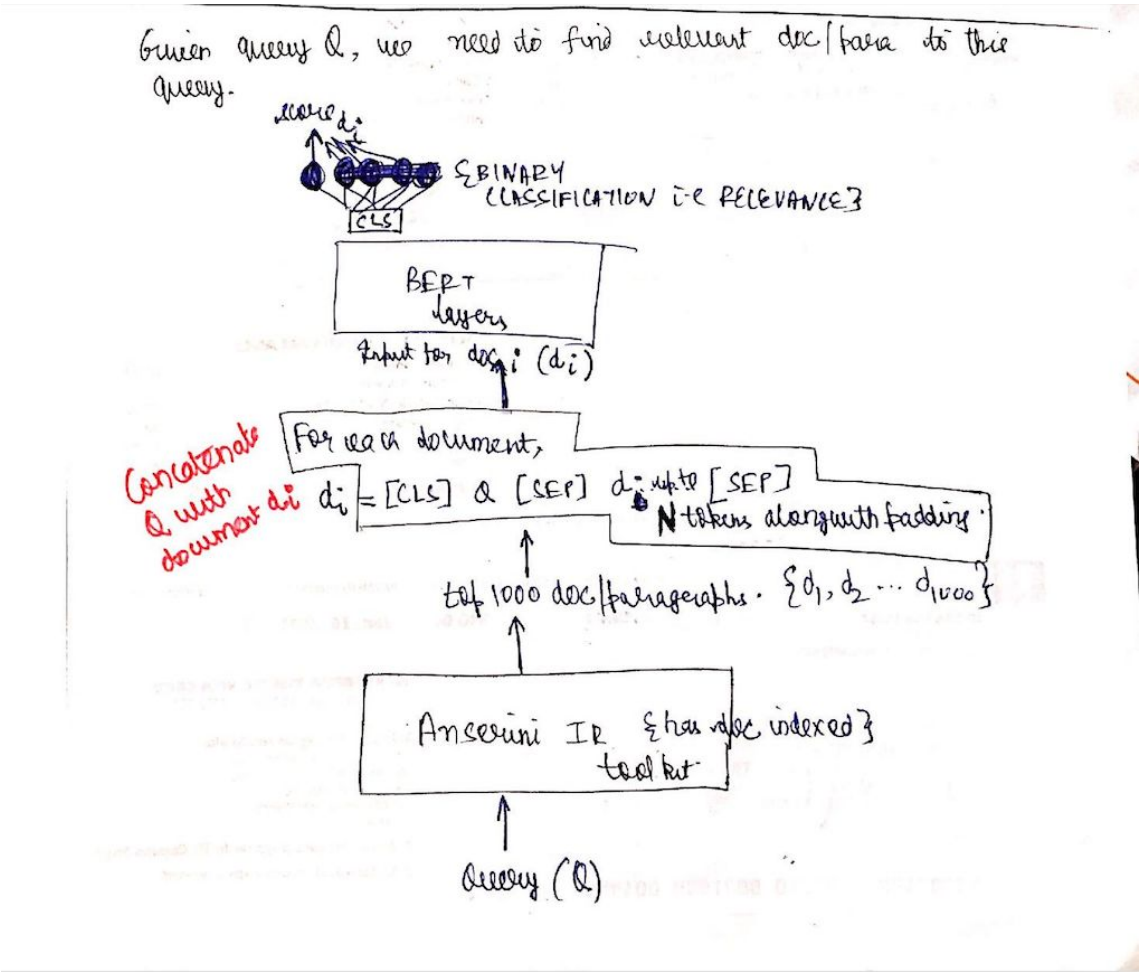
i. **Input to BERT**: Create a text sequence as **[[CLS], Q, [SEP], D, [SEP]].** This sequence is then padded to N tokens in a mini batch [N is maximum length in a batch] + document level relevance (y)

ii. **Output of BERT**: Output corresponding to [CLS] token is fed into a single layer NN for Binary Classification (relevant or not relevant)

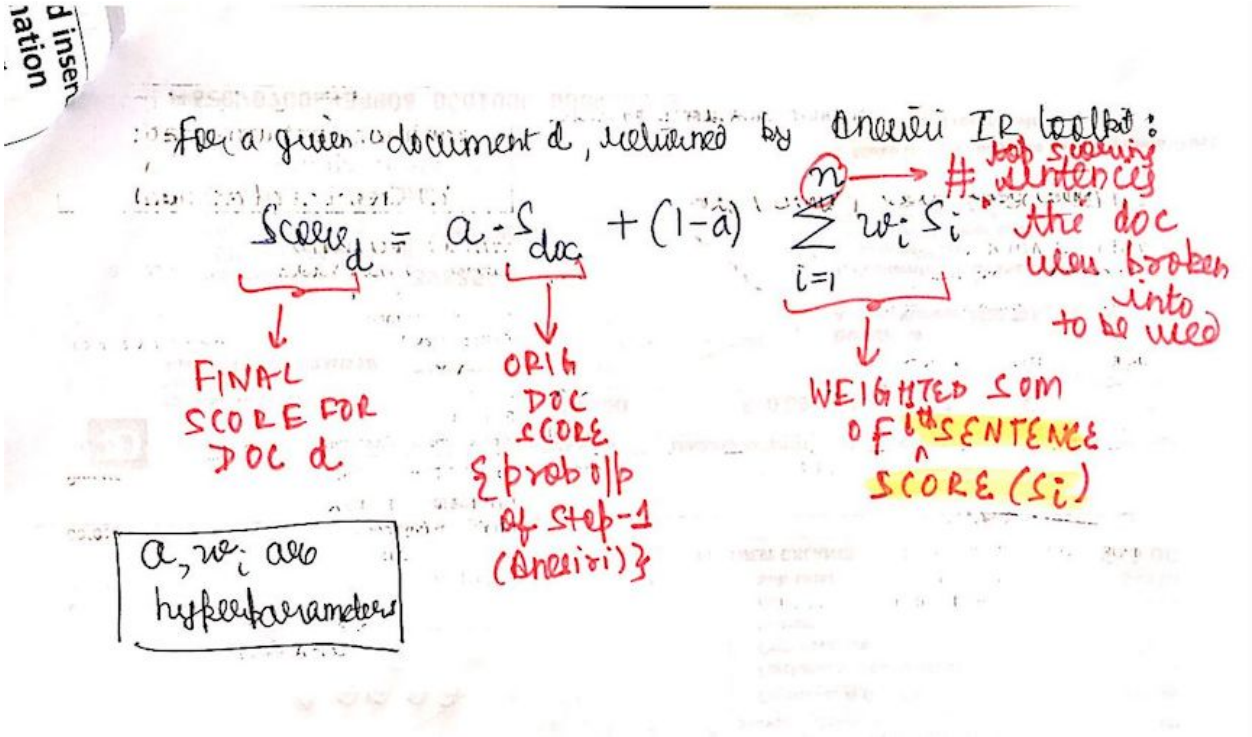
iii. **Fine Tuning BERT for score calculation/ relevance calculation:**

1. **Loss function**: Cross Entropy.
2. Fine Tuned on TREC Microblog Tracks + Union(TrecQA,WikiQA)
3. **Metric**: Average Precision (AP) and Precision at rank 30 (P30)

iv. Evaluation done on Robust04 newswire data



d. Handling documents spanning multiple lines (**total length of doc > MAX LENGTH BERT i.e. 512 tokens**)
   i. **IDEA**: Identify the **"best"** sentence in a document and use it as a proxy for document relevance.
   ii. Break a paragraph into component sentences. Pass each of the sentences through BERT, get a score for each sentence. Choose the sentence that has the **highest score** to be used as proxy of the document.



C. My Assessment/ Analysis:

a. A simple way to get document level relevance score by combining sentence level scores using LINEAR INTERPOLATION.
b. Gets SOTA performance on TREC dataset on metrics like MAP, Precision at rank 30.
c. No need to specifically fine tune BERT on **Doc Retrieval** task.
d. Good thing that code is available on GITHUB

D. LIMITATIONS/ CONFUSIONS for me

a. Simplistic way of extending sentence score to document score.
b. Not very sure how **Anserini IR toolkit** works and how it prioritizes which documents to return as relevant
c. Based on assumption that QA task and IR task are similar in terms of BERT training but in reality it is not. In other words, task differences between QA and document retrieval do not appear to hinder BERT's adaptability.

Name: Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval

The difference between the paper above and this is:

1. Uses BERT-Large model
2. Finetunes the BERT architecture on MS-MARCO, CAR and TREC Microblog datasets
3. Evaluates on Robust04, Core17 and Core 18 datasets
4. Metrics: Average Precision, P@20 and NDCG@20
5. **Conclusion**: first, that relevance models can be transferred quite straightforwardly across domains by BERT, and second, that effective document retrieval requires only "paying attention" to a small number of "top sentences" in each document.