UNICAMP

Article Presentation

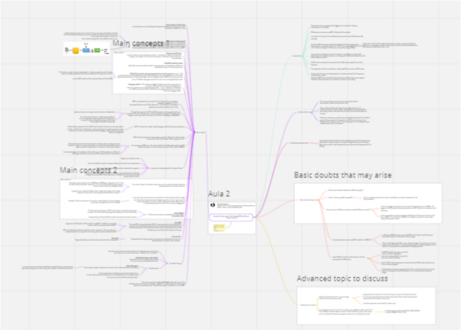
IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Student: Marcus Vinícius Borela de Castro

Pretrained Transformers for Text Ranking BERT and Beyond - Chapter 3 (partial)

By: Jimmy Lin, Rodrigo Nogueira, and Andrew Yates

Link



https://miro.com/app/board/uXjVO-OAf1w=/?moveToWidget=3458764548424671164&cot=14

miro

Main concepts - 1

Relevance classification

Convert the task into a text classification problem: to estimate the probability that each text belongs to the “relevant” class, and then at ranking (i.e., inference) time sort the texts by those estimates

Probability Ranking Principle

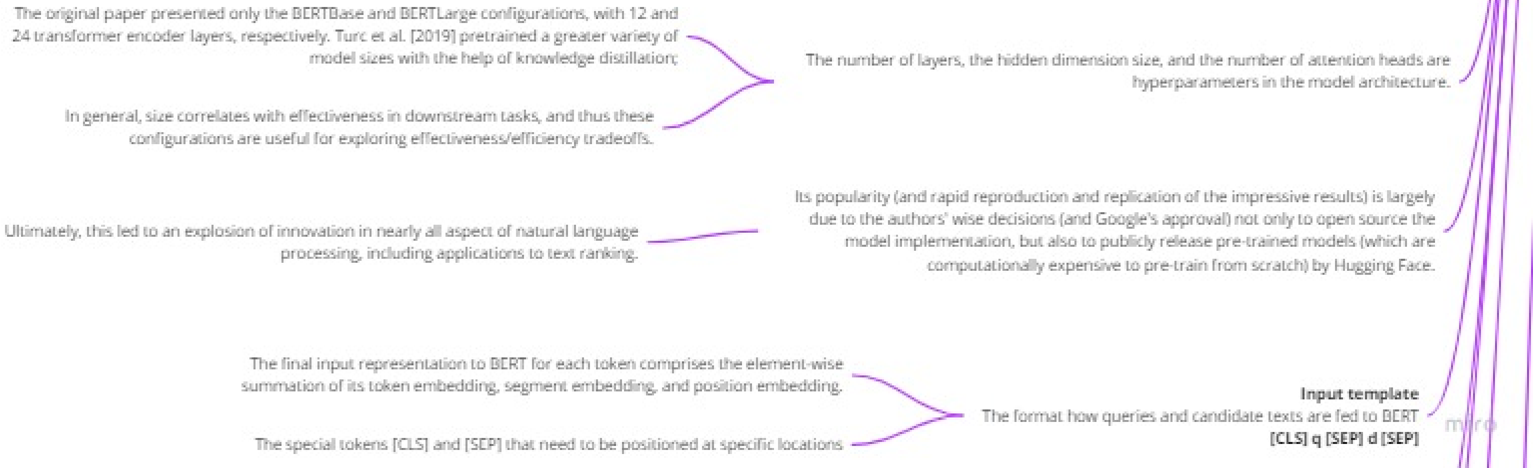
States that documents should be ranked in decreasing order of the estimated probability of relevance with respect to the information need

BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2019]

Is a neural network model for generating contextual embeddings for input sequences (which provide context-dependent representations of the input) in English, with a multilingual variant (“mBERT”) that can process input in over 100 different languages. Here we focus only on the monolingual English model.

A language model in NLP provides a probability distribution over arbitrary sequences of text tokens. BERT and GPT are often grouped together and referred to collectively as pretrained language models. . In truth, coaxing such probabilities out of BERT require a bit of effort, and transformers in general can do much more than “traditional” language models

Main concepts - 2



The original paper presented only the BERTBase and BERTLarge configurations, with 12 and 24 transformer encoder layers, respectively. Turc et al. [2019] pretrained a greater variety of model sizes with the help of knowledge distillation;

In general, size correlates with effectiveness in downstream tasks, and thus these configurations are useful for exploring effectiveness/efficiency tradeoffs.

The number of layers, the hidden dimension size, and the number of attention heads are hyperparameters in the model architecture.

Ultimately, this led to an explosion of innovation in nearly all aspect of natural language processing, including applications to text ranking.

Its popularity (and rapid reproduction and replication of the impressive results) is largely due to the authors' wise decisions (and Google's approval) not only to open source the model implementation, but also to publicly release pre-trained models (which are computationally expensive to pre-train from scratch) by Hugging Face.

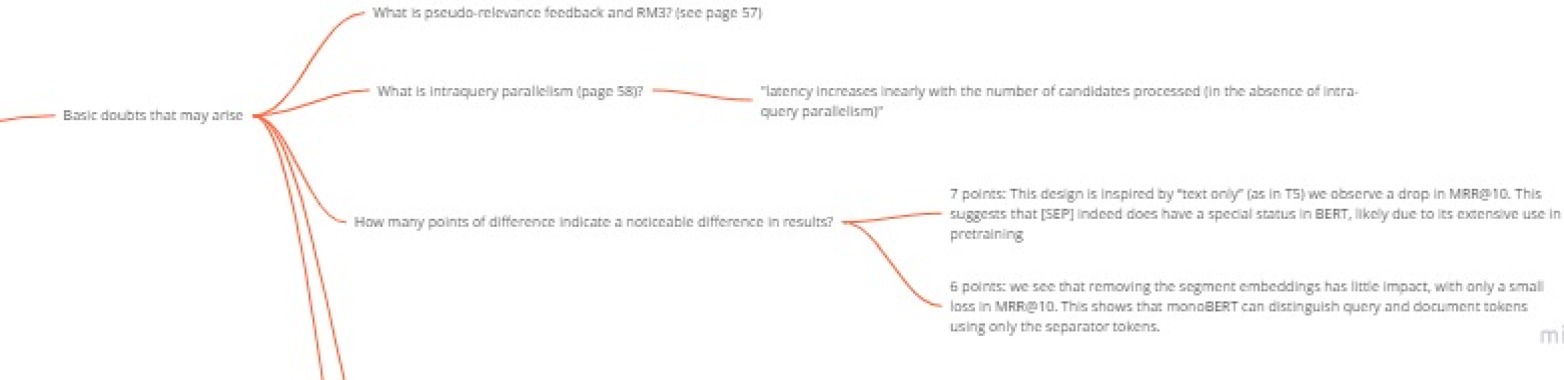
The final input representation to BERT for each token comprises the element-wise summation of its token embedding, segment embedding, and position embedding.

The special tokens [CLS] and [SEP] that need to be positioned at specific locations

Input template

The format how queries and candidate texts are fed to BERT

[CLS] q [SEP] d [SEP]



Basic doubts that may arise

What is pseudo-relevance feedback and RM3? (see page 57)

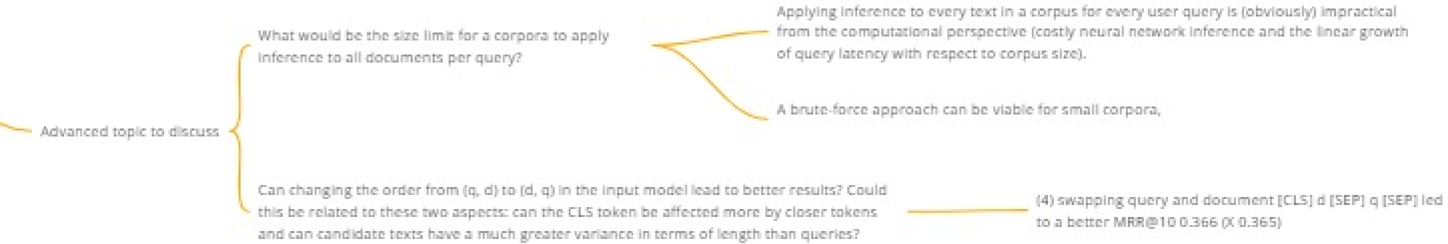
What is intraquery parallelism (page 58)?

"latency increases linearly with the number of candidates processed (in the absence of intra-query parallelism)"

How many points of difference indicate a noticeable difference in results?

7 points: This design is inspired by "text only" (as in T5) we observe a drop in MRR@10. This suggests that [SEP] indeed does have a special status in BERT, likely due to its extensive use in pretraining

6 points: we see that removing the segment embeddings has little impact, with only a small loss in MRR@10. This shows that monoBERT can distinguish query and document tokens using only the separator tokens.



Advanced topic to discuss

What would be the size limit for a corpora to apply inference to all documents per query?

Applying inference to every text in a corpus for every user query is (obviously) impractical from the computational perspective (costly neural network inference and the linear growth of query latency with respect to corpus size).

A brute-force approach can be viable for small corpora,

Can changing the order from (q, d) to (d, q) in the input model lead to better results? Could this be related to these two aspects: can the CLS token be affected more by closer tokens and can candidate texts have a much greater variance in terms of length than queries?

(4) swapping query and document [CLS] d [SEP] q [SEP] led to a better MRR@10 0.366 (X 0.365)