Pretrained Transformers for Text Ranking: BERT and Beyond

•••

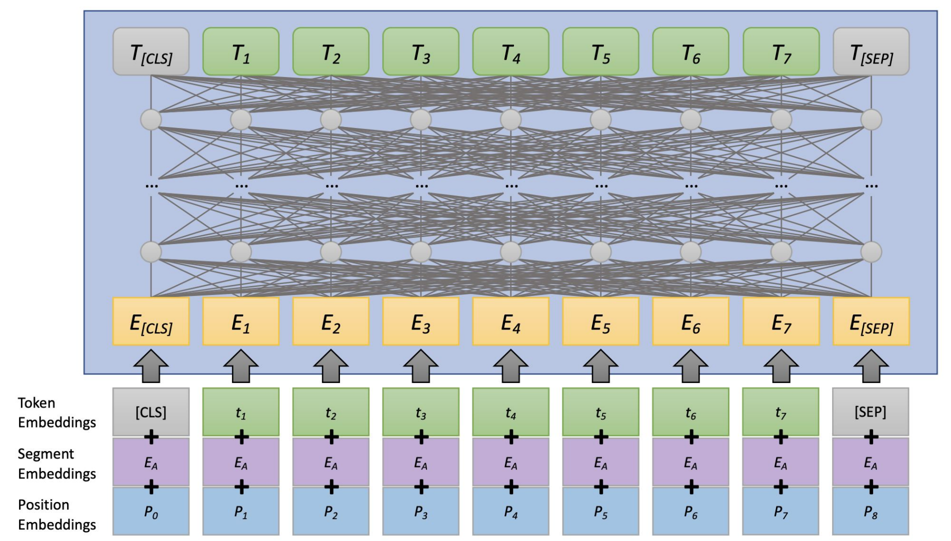
16 de março de 2023

Main concepts

- Text ranking can be modeled as a text classification problem, and the texts are to be ranked based on the probability that each item belongs to the desired class.
- **Probability Ranking Principle**
  - Training a classifier 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.

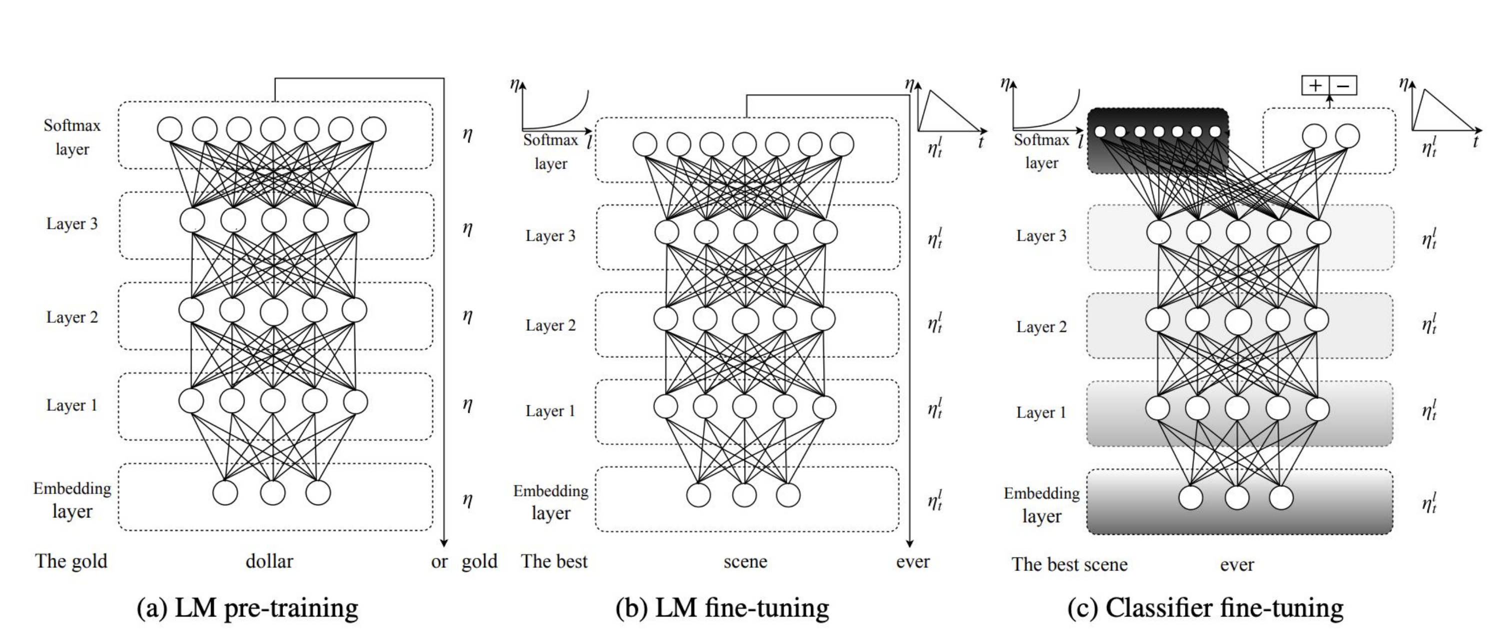
Main concepts - BERT

- Is a neural network model for generating contextual embeddings for input sequences in English.
- BERT takes as input a sequence of tokens and outputs a sequence of contextual embeddings, which provide context-dependent representations.
- BERT introduced the concept of "masked language model" (MLM) pretraining objective.


$T_{[CLS]}$ T_1 T_2 T_3 T_4 T_5 T_6 T_7 $T_{[SEP]}$
$E_{[CLS]}$ E_1 E_2 E_3 E_4 E_5 E_6 E_7 $E_{[SEP]}$
Token Embeddings
[CLS] t_1 t_2 t_3 t_4 t_5 t_6 t_7 [SEP]
Segment Embeddings
E_A
Position Embeddings
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8

Main concepts - BERT

The idea of pretraining has a long history. ULMFiT (Universal Language Model Fine-tuning) likely deserves the credit for popularizing the concept of pretraining using language modeling objectives and then fine-tuning on task-specific data.



(a) LM pre-training

(b) LM fine-tuning

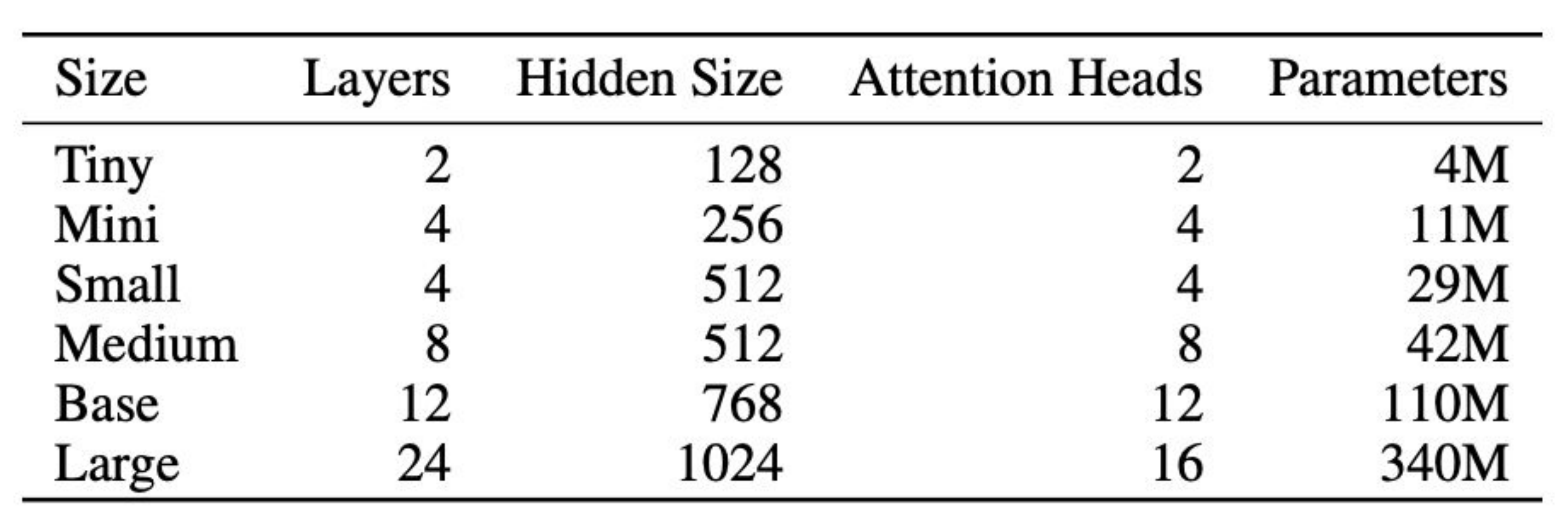
(c) Classifier fine-tuning

Main concepts - BERT

- Input sequences to BERT are usually tokenized with the WordPiece tokenizer, although BPE (Byte Pair Encoding) is a common alternative.
- These tokenizers have the aim of reducing the vocabulary space by splitting words into “subwords”.

Main concepts - BERT

The original paper presented only the BERTBase and BERTLarge configurations, with 12 and 24 transformer encoder layers. Afterward, a greater variety of model sizes was trained with the help of knowledge distillation.



Size	Layers	Hidden Size	Attention Heads	Parameters
Tiny	2	128	2	4M
Mini	4	256	4	11M
Small	4	512	4	29M
Medium	8	512	8	42M
Base	12	768	12	110M
Large	24	1024	16	340M

Simple Relevance Classification: monoBERT

The task of relevance classification is to estimate a score ***s_i*** quantifying how relevant a candidate text ***d_i*** is to a query ***q***

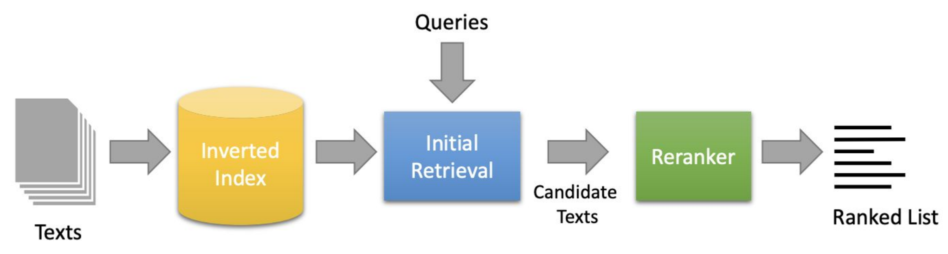
$$P(\text{Relevant} = 1 | d_i, q)$$

Retrieve and rerank architecture

- Candidate texts are identified from the corpus using keyword search, usually with bag-of-words queries against inverted indexes

- Ordered by a scoring function based on exact term matches such as BM25

- BERT inference is then applied to rerank these candidates to generate a score



Obrigado

Manoel Veríssimo