Comparing Pointwise, Pairwise and Listwise Ranking Approaches On BERT-based Cross-Encoders

Luis Schulte
Heidelberg University
schulte@cl.uni-heidelberg.de

Abstract

Ranking has been a fundamental task in information retrieval for many decades. In recent years, re-ranking became a critical component of retrieval-augmented generation (RAG) systems, where selecting the most relevant passages significantly impacts downstream performance of multi-stage retrieval pipelines.
In this university project, I implement three different BERT-based cross-encoder models for ranking and systematically compare three training paradigms: *pointwise*, *pairwise*, and *listwise*.

1 Introduction

Modern IR pipelines, including retrieval-augmented generation (RAG) systems, often use a two-stage approach: a fast first-stage retriever (e.g., BM25 or a bi-encoder) surfaces a broader candidate set (often more than 100 retrieved documents or passages). Then a more accurate but computationally expensive, and hence slower cross-encoder re-ranks the candidates by scoring each query–passage pair. Cross-encoders jointly encode query and passage as a single sequence. This allows full cross-attention of transformer models, which greatly improves the interaction between query and passage representations. As a result, cross-encoders are usually computationally expensive, but vastly superior in effectiveness. In practice, a bi-encoder is used in conjunction with a tiny classification head to train a cross-encoder ranking model. (Zhang et al., 2023)

A few different ranking strategies have been proposed in the literature. A common classification is to group these ranking approaches into three distinct categories: *pointwise*, *pairwise* and *listwise* ranking. (Liu, 2009)

Task: This is identical for each of the three methods: Given a query q and a set of candidate passages $C = (c_1, ..., c_n)$, the model function $f(q, C)$ is supposed to return a permutation of C that ranks each passage c_i by its relevance to q.

Method: Depending on the ranking approach, the method varies.

- **Pointwise** models concatenate the query with each candidate passage: `[CLS] query [SEP] passage [SEP]`. The resulting sequence is fed into the cross-encoder, outputting a single logit per query–passage pair. The model is trained on this binary task to minimize the error betwee predicted and true relevance scores. A typical loss function is binary cross-entropy loss.
- **Pairwise** models concatenate the query with two candidate passages: `[CLS] query [SEP] passage_a [SEP] passage_b [SEP]`. The resulting sequence is fed into the cross-encoder, outputting two logits, one for each passage. The model is trained on this binary task to minimize the error between the predicted and true relevance scores of the two passages. A typical loss function is pairwise loss, such as hinge loss or RankNet loss.
- **Listwise** models concatenate the query with all candidate passages: `[CLS] query [SEP] passage_1 [SEP] passage_2 [SEP] ... [SEP] passage_n [SEP]`. The resulting sequence is fed into the cross-encoder, outputting a relevance score for each candidate passage in a single forward pass. The model is trained to minimize the error between the predicted and true relevance scores for all candidate passages. A typical loss function is listwise loss, such as ListNet loss or ListMLE loss.

For all three approaches, a probability distribution over the candidate passages is obtained by applying a softmax function to the output logits. The passages are then ranked based on these probabilities.

2 Methodology

2.1 Model Architecture

Three different cross-encoder architectures are implemented, one for each ranking approach. Ideally, a single bi-encoder is used as the foundation model, with a small classification or regression head on top to adapt to the specific ranking approach. In this case, BERT (Devlin et al., 2018) is preferred due to its wide adoption, flexibility, and strong performance on many NLP tasks.

Listwise ranking requires the query and all candidate passages to be processed in a single forward pass. The resulting input sequence exceeds the maximum token limit of BERT (512 tokens) in most cases. Therefore, a similiar model with a greater context size is used to implement a listwise ranking cross-encoder. Specifically Longformer (Beltagy et al., 2020), a RoBERTa-based model, is used.

- `CrossEncoderBERT` is the BERT-based cross-encoder used for pointwise and pairwise ranking. A variable classfication head allows to adapt to either ranking approach. A small multi-layer perceptron (MLP) is used for classification, with a single output unit, representing the relevance score in the case of pointwise ranking, and two output units in the case of pairwise ranking, where each logit represents the score of one of the two input passages.
- `CrossEncoderLongformer` is the Longformer-based cross-encoder used for listwise ranking. It uses a linear layer to predict relevance scores for each candidate passage in the input sequence. The same MLP architecture is applied to the foundation model, although the output layer has a fixed-size output, accounting for enough candidate passages to fit into the model's maximum input size. Missing candidates are automatically padded. Special separator tokens (e.g., `[CAND]`) are inserted in order to later pool the output scores correctly.

2.2 Training

Each model is trained on the same dataset with the a similiar training setup to ensure comparability.

3 Experimental Setup

3.1 Data

MS MARCO (*Microsoft MAchine Reading Comprehension*) is a popular IR dataset that is used for many tasks, including ranking. It features real-world user queries and human-supervised candidate answers with relevance labels. It comes pre-splitted with a training, validation and test configuration (80/10/10), all together encompassing more than one million entries (Nguyen et al., 2016).

mMARCO is a multilingual variant of MS MARCO (Bonifacio et al., 2021). It does not include a pre-splitted configuration, so samples have been assigned based on an 80/10/10 split for training, validation and testing respectively. The assignment was done by hashing each sample and comparing this uniform distribution to the split ratios.

3.2 Evaluation Metrics

For evaluation, common IR rankin metrics are applied.

- Mean Reciprocal Rank (MRR), which measures the average of the reciprocal ranks of the first relevant document for a set of queries. It is particularly useful when the focus is on retrieving at least one relevant document quickly (Craswell, 2009).
- ListNet, a listwise ranking metric that evaluates the quality of a ranked list by comparing the predicted ranking to the ground truth relevance scores. It is based on the probability distribution over permutations of the ranked list Essentially, this is the cross-entropy loss (CEL) between the predicted and true relevance distributions (Cao et al., 2007).

4 Results & Dicussion

The inital objective to compare all three ranking approaches by training and evaluating all three models on the same dataset could not be achieved in its entirety.

Due to time constraints and inherent issues with the models and dataset, only the pointwise and listwise ranking models could be trained and partially evaluated.

Specifically, significant performance issues during training prevented the listwise model from converging to a satisfactory level. This renders

a direct inter-model comparison infeasible and meaningless.

Pointwise and pairwise approaches are usually outperformed by listwise approaches (Tax et al., 2015). However, this may not hold for each scenario, especially when considering efficiency aspects such as in high-throughput settings.

Another, more recent approach to listwise ranking is to use GPT-style generative models where the model generates the most relevant passage given a query and a set of candidate passages. (Zhang et al., 2023)

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150*.

Luiz Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Frassetto Nogueira. 2021. mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset. *CoRR*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In volume 227, pages 129–136.

Nick Craswell. 2009. Mean Reciprocal Rank. In *Encyclopedia of Database Systems*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*.

Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CoRR*.

Niek Tax, Sander Bockting, and Djoerd Hiemstra. 2015. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management*, 51(6):757–772.

Xinyu Zhang, Sebastian Hofstätter, Patrick Lewis, Raphael Tang, and Jimmy Lin. 2023. Rank-without-GPT: Building GPT-Independent Listwise Rerankers on Open-Source Large Language Models.