Searching for Solutions to Puzzles & Riddles
Part 2: Neural Networks

Ellis Fitzgerald

COS 470

November 18 2024

Introduction

Models Chosen:

- SBERT BiEncoder: msmarco-distilbert-base-v4
- SBERT CrossEncoder: ms-marco-MiniLM-L-6-v2

Document Set:

- Puzzling Stack Exchange snapshot: 63,997 documents.
- answers.json
- topics_1.json w/ respective qrel_1.tsv
- topics_2.json

Dependencies/Citations:

- NLTK
- BeautifulSoup4
- SBERT
- Ranx
- Pandas
- Numpy
- Torch
- MS (Microsoft) Marco

Implementation

- Preprocessing / Text
  - Clean HTML, code, unicode, punctuation
  - Topics = Title + Body + Tags
  - Answer = Text
- Justification for Models:
  - Both models are trained on MS Marco.
  - MS Marco: Dataset from Microsoft of 500k real queries matched with relevant passages.
- Training Arguments
  - Bi-Encoder & Cross-Encoder Epochs: 1 and 4 respectively
  - Bi-Encoder Loss: CoSENTLoss
  - Bi-Encoder & Cross-Encoder Evaluators: InformationRetrievalEvaluator and CERerankingEvaluator respectively

Initial Concerns & Assumptions

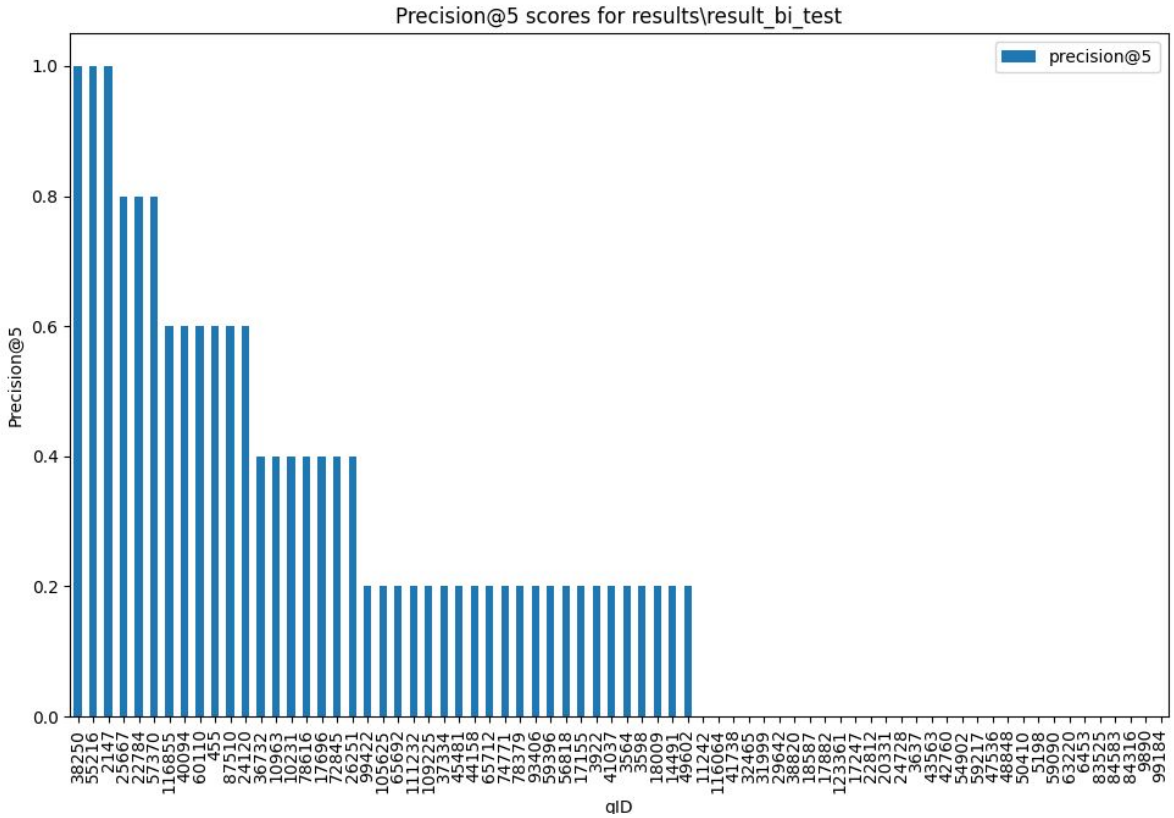
Concerns:

- Overfitting with qrel split (train, evaluation, and test)
- PyTerrier's BM25 = ***solid***
- BERT-based models are 512 tokens, which corresponds to about 300-400 words (for English)
- Text longer than that will be truncated... It might not work well for longer text.
- The longest Answer len: 26028.
- Does having both models pretrained on MS Marco have any benefit?

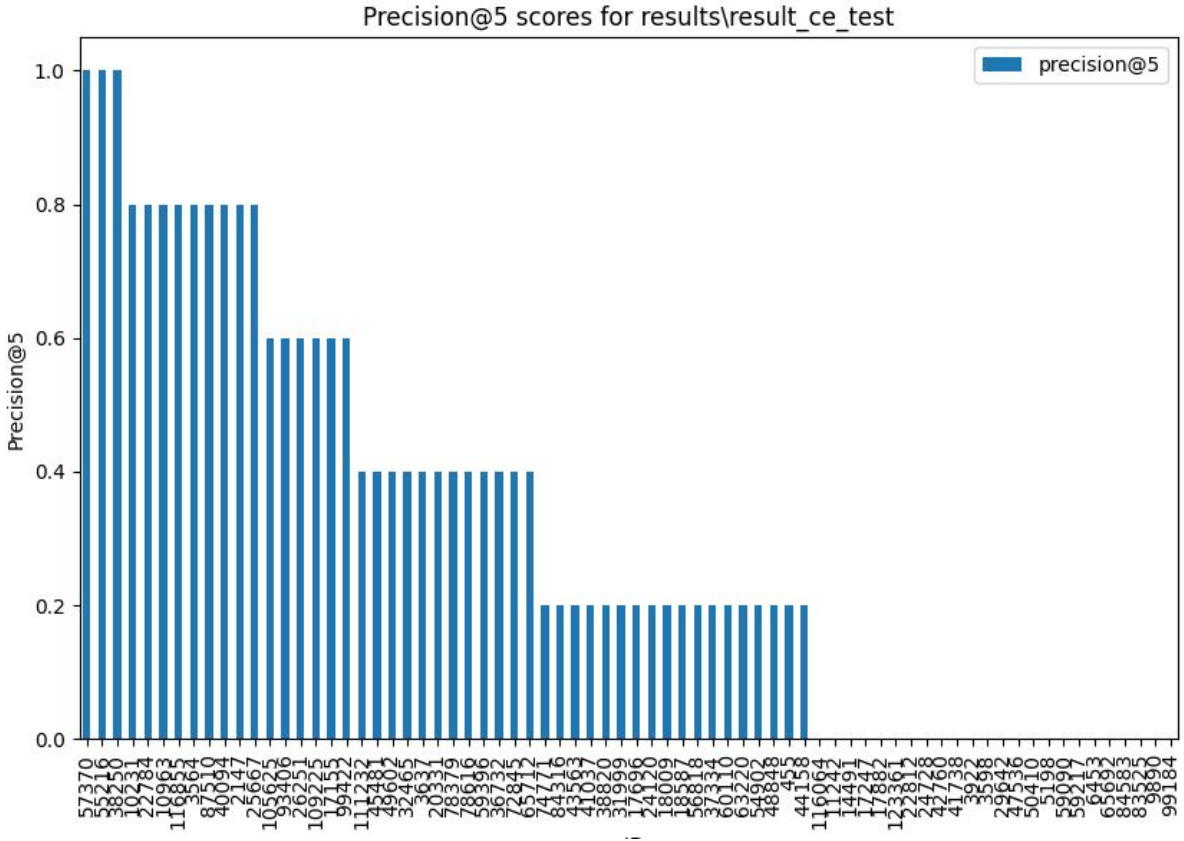
Assumptions:

- Cross-Encoder will perform better

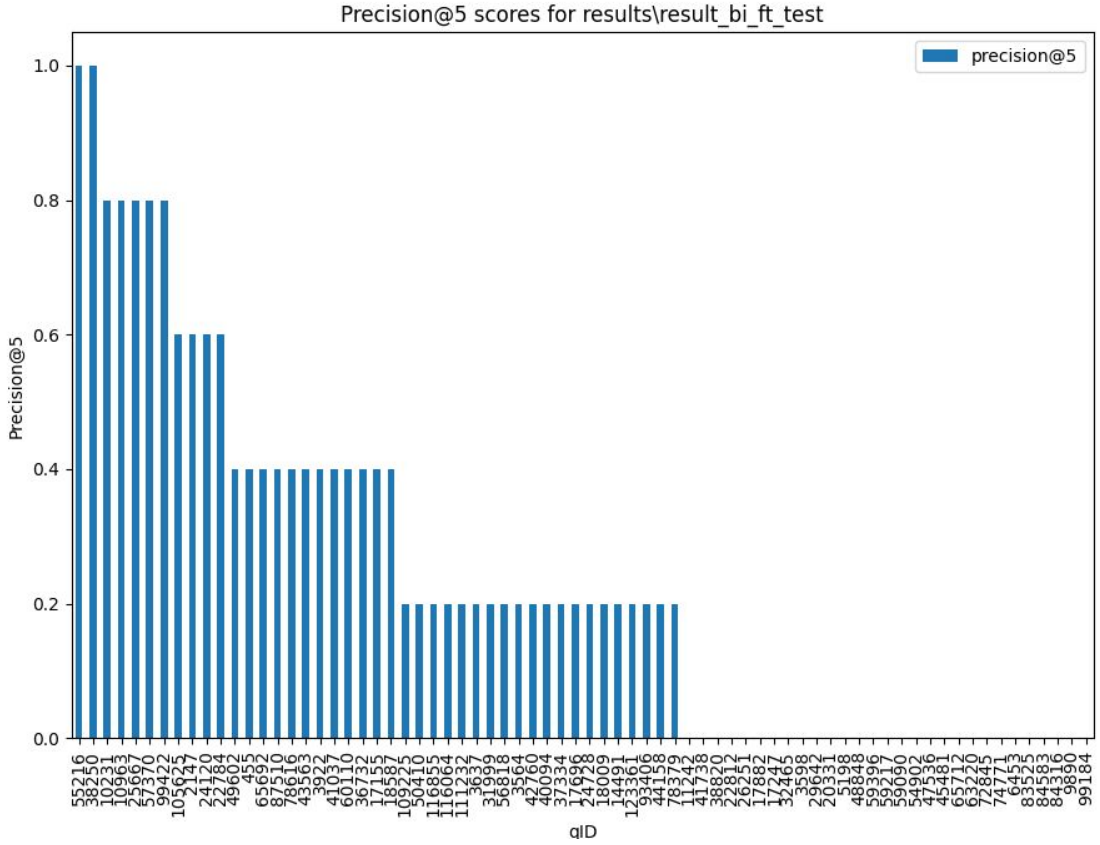
Experimental Analysis



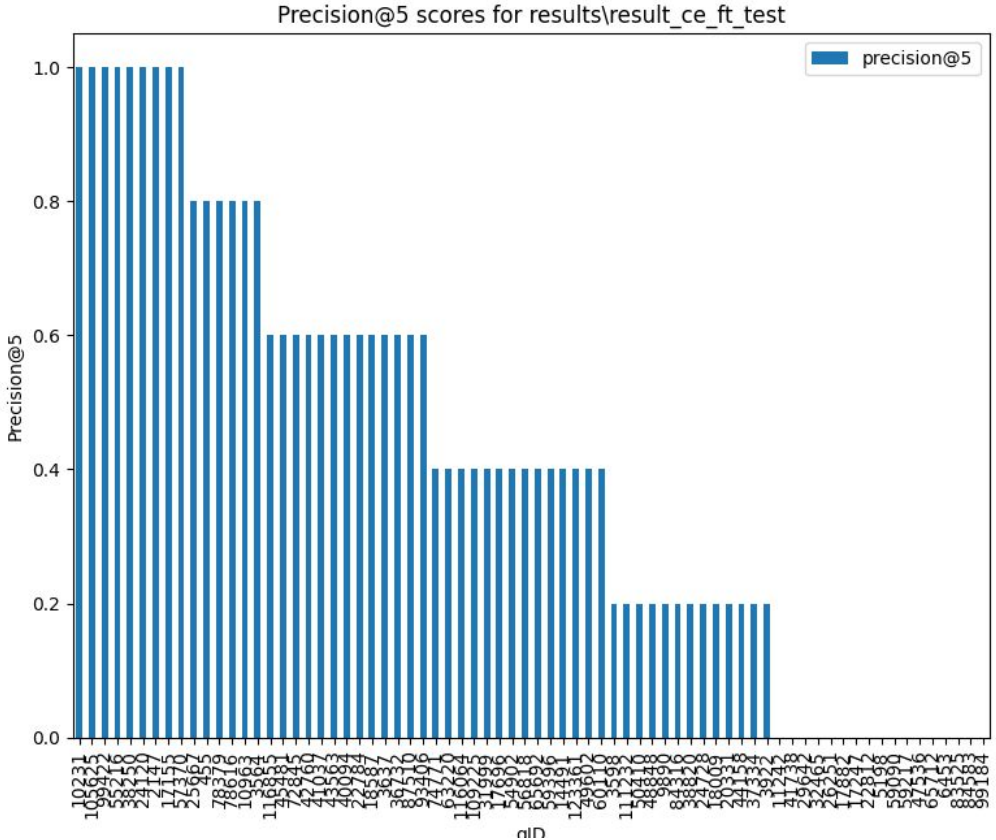
Precision@5 for Pretrained Bi-Encoder



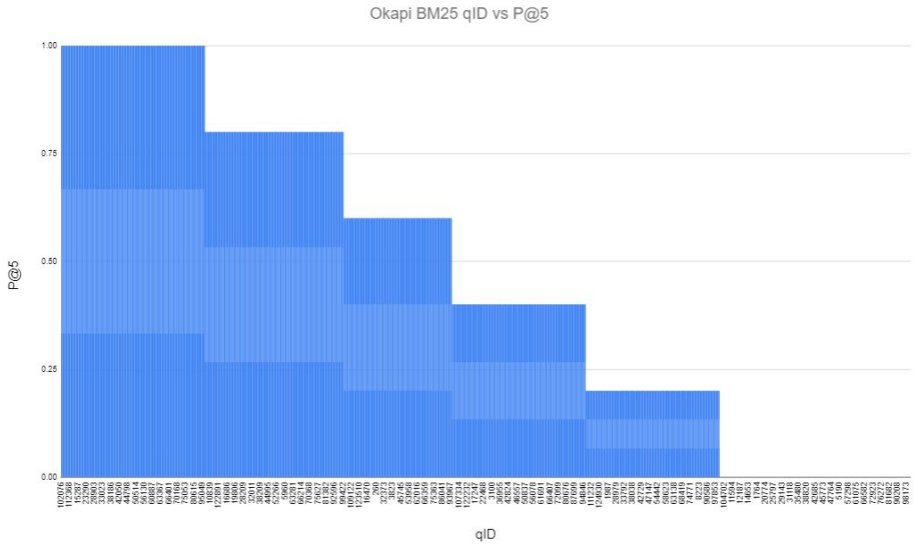
Precision@5 for Pretrained Cross-Encoder

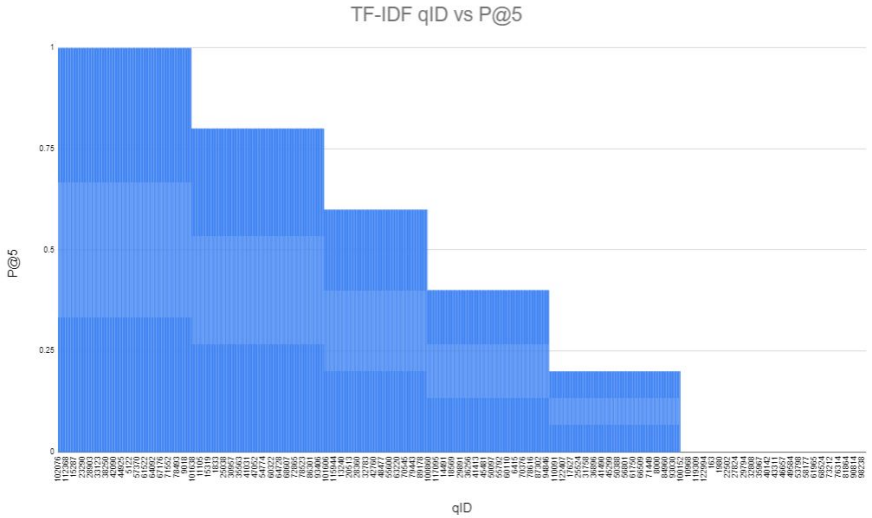


Precision@5 for Fine-Tuned Bi-Encoder

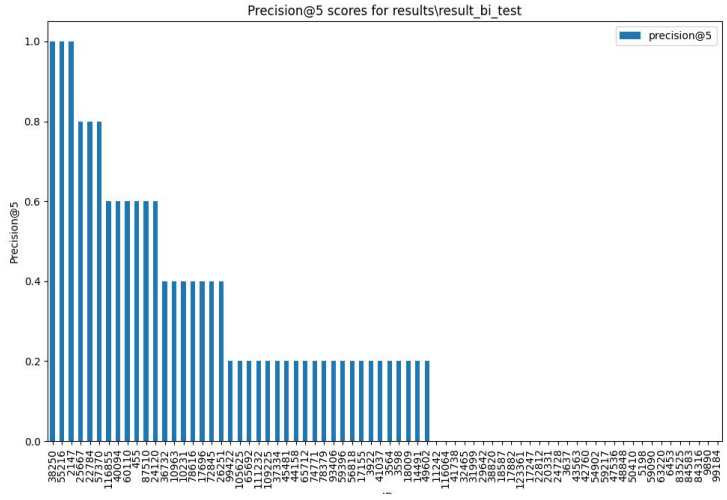


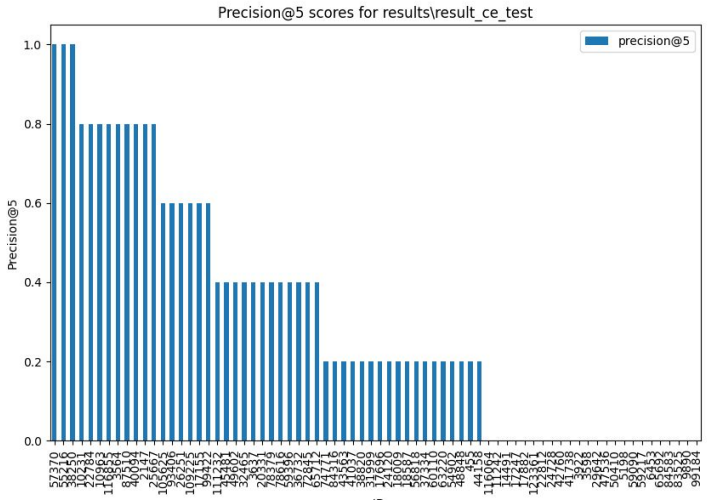
Precision@5 for Fine-Tuned Cross-Encoder

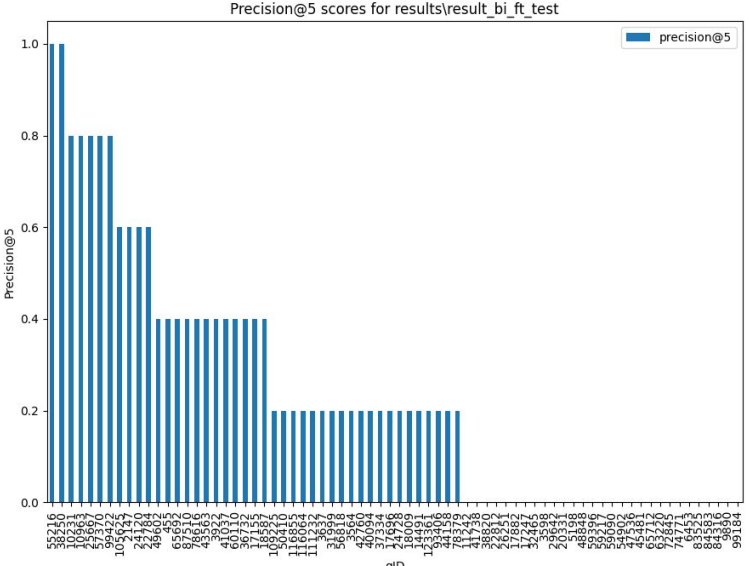


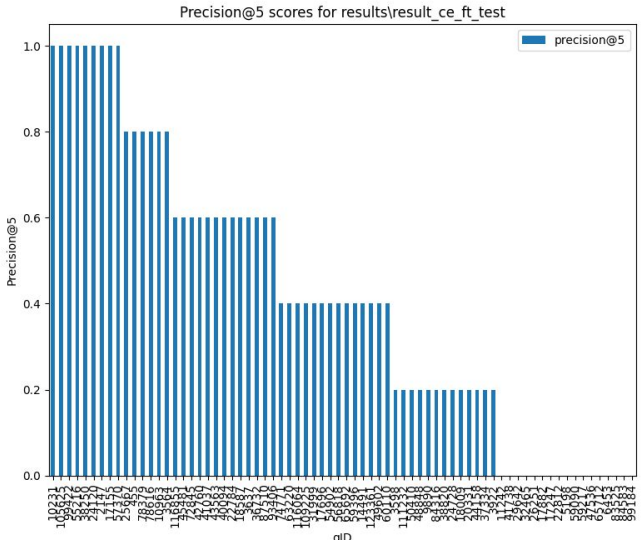


BM25 & TF-IDF Result Graph P@5


Precision@5 scores for results\result_bi_test
precision@5
Precision@5
1.0
0.8
0.6
0.4
0.2
0.0


Precision@5 scores for results\result_ce_test
precision@5
Precision@5
1.0
0.8
0.6
0.4
0.2
0.0


Precision@5 scores for results\result_bi_ft_test
precision@5
Precision@5
1.0
0.8
0.6
0.4
0.2
0.0


Precision@5 scores for results\result_ce_ft_test
precision@5
Precision@5
1.0
0.8
0.6
0.4
0.2
0.0

Good Results

Pretrained Bi-Encoder

Q: 38250
I'm generous if you like me, but greedy if you hate me
A: 38276
"You are a king"
Score: 1.0

Pretrained Cross-Encoder

Q: 57370
Who killed Jeremy? (Peter, Tom, John, Ralph...)
A: 57385
(Peter, Tom, John, Ralph...)
Score: 1.0

Fine-tuned Bi-Encoder

Q: 55216
Batman vs 4 villains
A: 55235
(Riddler, Penguin, Joker, Two-Face...)
Score: 1.0

Fine-tuned Cross-Encoder

Q: 10231
Don't be Sexist - What am I?
A: 10307
"The answer may be a human ovary"
Score: 1.0

Uniform Bad Result

Q: 99184

All wrapped in functions...(math containers)

A: 109108

Unrelated algebraic proof...

Score: 0.0

Table 5. Overall effectiveness of the models. The best results are highlighted in boldface. Superscripts denote significant differences in paired Student's t-test with $p \leq 0.01$.

#	Model	NDCG@5	NDCG@10	P@5	P@10	MAP	BPref	MRR
a	results_bi_test	0.236	0.240	0.225	0.147	0.189	nan	0.463
b	results_ce_test	0.335^{a}	0.337^{a}	0.308^{a}	0.203^{a}	0.268^{a}	nan	0.566
c	results_bi_ft_test	0.271	0.285	0.239	0.169	0.228	nan	0.474
d	results_ce_ft_test	$\mathbf{0.428}^{abc}$	$\mathbf{0.435}^{abc}$	$\mathbf{0.414}^{abc}$	$\mathbf{0.268}^{abc}$	$\mathbf{0.366}^{abc}$	nan	$\mathbf{0.641}^{ac}$

Significance Test

Table 5. Overall effectiveness of the models. The best results are highlighted in boldface. Superscripts denote significant differences in paired Student's t-test with $p \leq 0.01$.

#	Model	NDCG@5	NDCG@10	P@5	P@10	MAP	BPref	MRR
a	results_bi_test	0.236	0.240	0.225	0.147	0.189	nan	0.463
b	results_ce_test	0.335^{a}	0.337^{a}	0.308^{a}	0.203^{a}	0.268^{a}	nan	0.566
c	results_bi_ft_test	0.271	0.285	0.239	0.169	0.228	nan	0.474
d	results_ce_ft_test	$\mathbf{0.428}^{abc}$	$\mathbf{0.435}^{abc}$	$\mathbf{0.414}^{abc}$	$\mathbf{0.268}^{abc}$	$\mathbf{0.366}^{abc}$	nan	$\mathbf{0.641}^{ac}$

name	map	ndcg	bpref	recip_rank	mrt	P@1	P@5	P@10	P@100
BM25	0.45917	0.54013	0.52513	0.68828	13.44275	0.61003	0.47047	0.31365	0.04139
TF-IDF	0.45419	0.53544	0.5208	0.68659	13.90481	0.60864	0.46546	0.30926	0.04107

Significance Test (Difference in format from PyTerrier Experiment & Ranx Compare)

BM25 is still superior (and/or my implementation is just not up to par)

Table 5. Overall effectiveness of the models. The best results are highlighted in boldface. Superscripts denote significant differences in paired Student's t-test with $p \leq 0.01$.

#	Model	NDCG@5	NDCG@10	P@5	P@10	MAP	BPref	MRR
a	results_bi_test	0.236	0.240	0.225	0.147	0.189	nan	0.463
b	results_ce_test	0.335^{a}	0.337^{a}	0.308^{a}	0.203^{a}	0.268^{a}	nan	0.566
c	results_bi_ft_test	0.271	0.285	0.239	0.169	0.228	nan	0.474
d	results_ce_ft_test	$\mathbf{0.428}^{abc}$	$\mathbf{0.435}^{abc}$	$\mathbf{0.414}^{abc}$	$\mathbf{0.268}^{abc}$	$\mathbf{0.366}^{abc}$	nan	$\mathbf{0.641}^{ac}$

name	map	ndcg	bpref	recip_rank	mrt	P@1	P@5	P@10	P@100
BM25	0.45917	0.54013	0.52513	0.68828	13.44275	0.61003	0.47047	0.31365	0.04139
TF-IDF	0.45419	0.53544	0.5208	0.68659	13.90481	0.60864	0.46546	0.30926	0.04107

Significance Test (Difference in format from PyTerrier Experiment & Ranx Compare)

Conclusion

Observations:

- Different top results for each of the models
- Maybe fusing those results would be beneficial?

Suggestions for next time:

- Approximate Nearest Neighbor (ANN) Search (not as exact, but faster and interesting concept) FAISS, Annoy
- Explore other NN not in the SBERT family: Gensim FastText

Showcase

Github

User: EllisFitzUSM

Repo: PuzzlesNeuralNetworkIR

https://github.com/EllisFitzUSM/PuzzlesNeuralNetworkIR