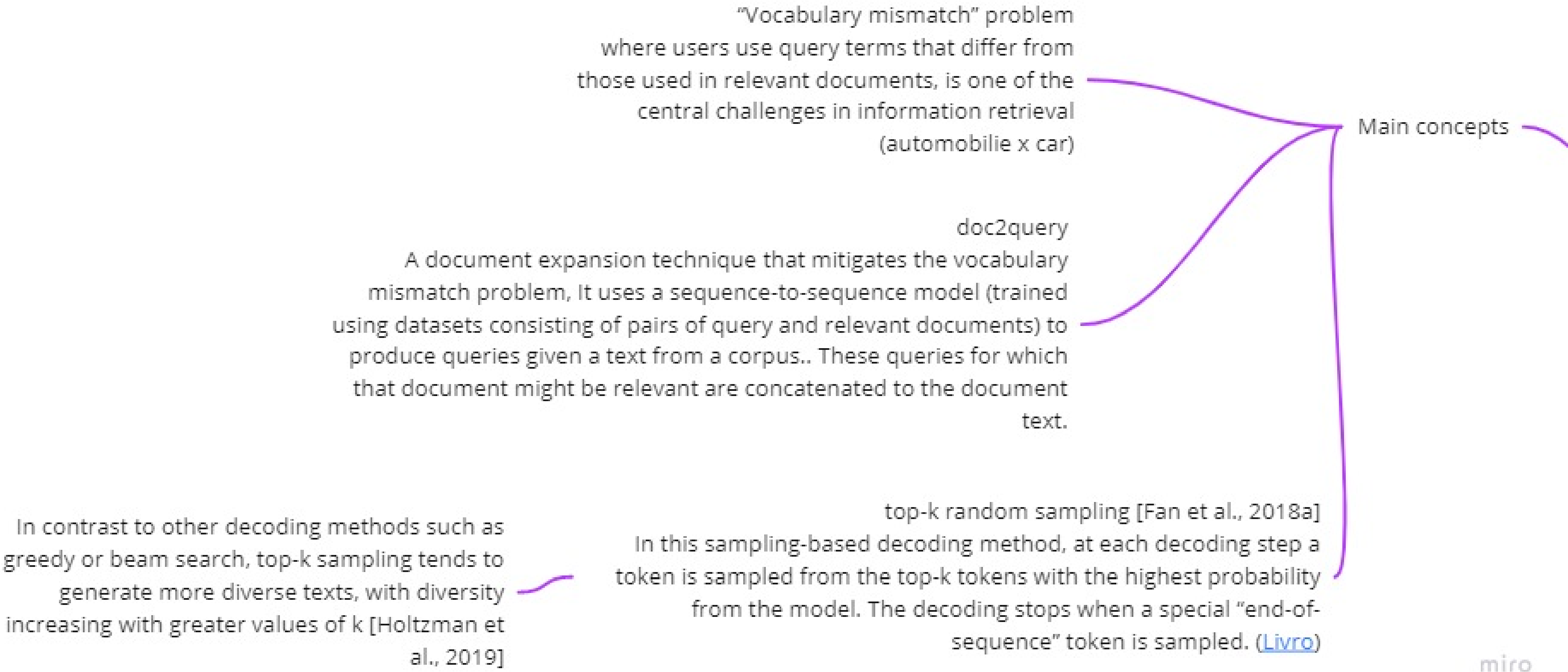
UNICAMP

Article Presentation

IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Student: Marcus Vinícius Borela de Castro

Document Expansion by Query Prediction From doc2query to docTTTTTquery



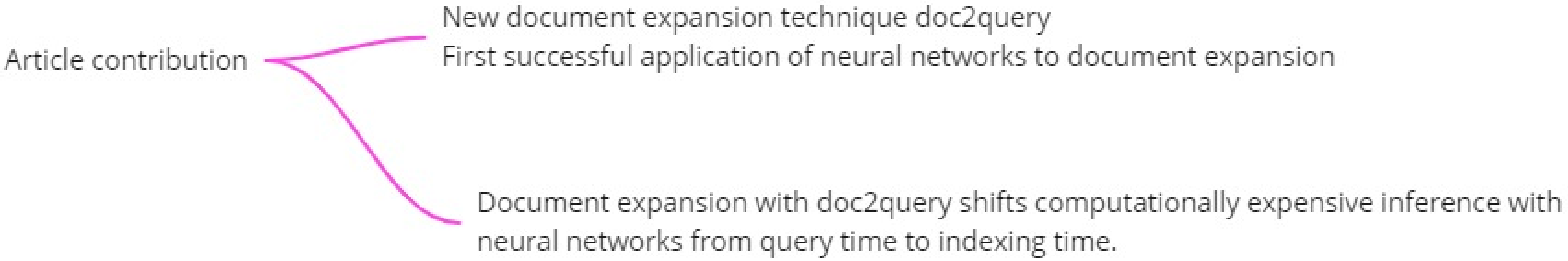
Main concepts

"Vocabulary mismatch" problem
where users use query terms that differ from those used in relevant documents, is one of the central challenges in information retrieval
(automobilie x car)

doc2query
A document expansion technique that mitigates the vocabulary mismatch problem, It uses a sequence-to-sequence model (trained using datasets consisting of pairs of query and relevant documents) to produce queries given a text from a corpus.. These queries for which that document might be relevant are concatenated to the document text.

top-k random sampling [Fan et al., 2018a]
In this sampling-based decoding method, at each decoding step a token is sampled from the top-k tokens with the highest probability from the model. The decoding stops when a special "end-of-sequence" token is sampled. (Livro)

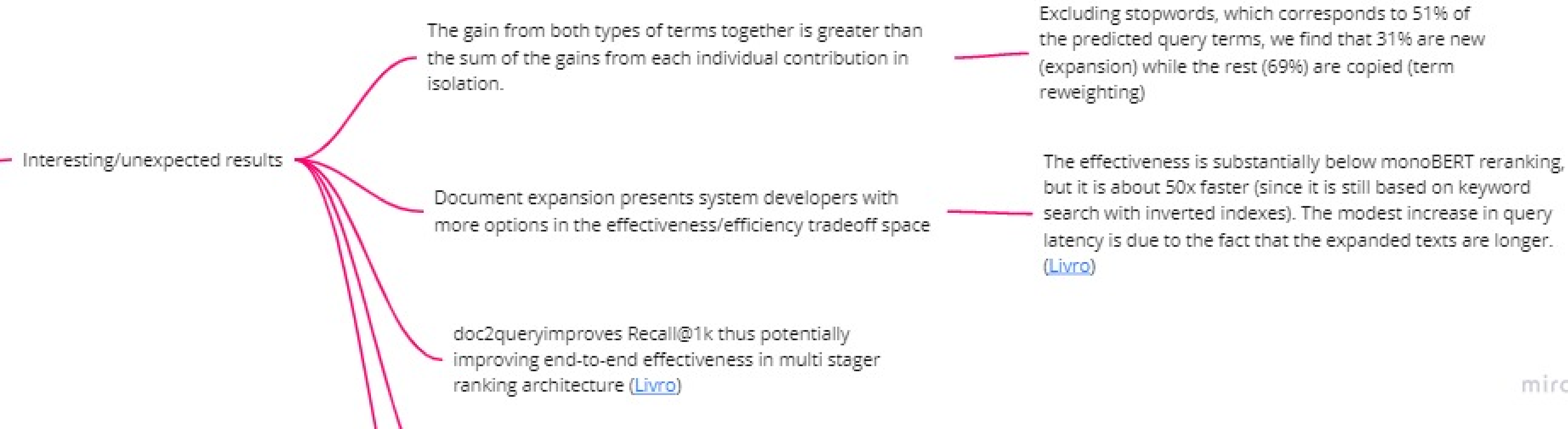
In contrast to other decoding methods such as greedy or beam search, top-k sampling tends to generate more diverse texts, with diversity increasing with greater values of k [Holtzman et al., 2019]



Article contribution

New document expansion technique doc2query
First successful application of neural networks to document expansion

Document expansion with doc2query shifts computationally expensive inference with neural networks from query time to indexing time.



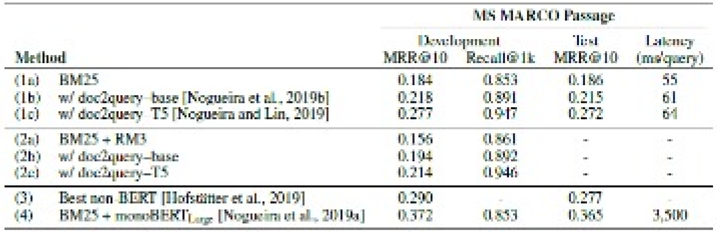
Interesting/unexpected results

- The gain from both types of terms together is greater than the sum of the gains from each individual contribution in isolation.
  - Excluding stopwords, which corresponds to 51% of the predicted query terms, we find that 31% are new (expansion) while the rest (69%) are copied (term reweighting)
- Document expansion presents system developers with more options in the effectiveness/efficiency tradeoff space
  - The effectiveness is substantially below monoBERT reranking, but it is about 50x faster (since it is still based on keyword search with inverted indexes). The modest increase in query latency is due to the fact that the expanded texts are longer. (Livro)
- doc2queryimproves Recall@1k thus potentially improving end-to-end effectiveness in multi stager ranking architecture (Livro)

miro

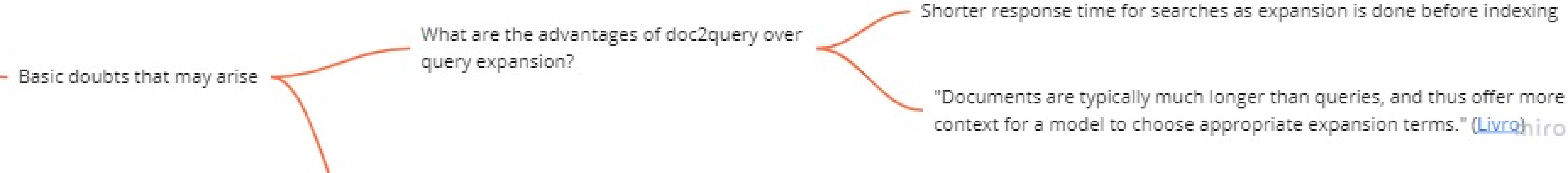
It appears clear that pretraining makes the crucial difference (1b x 1c) as even the T5-small model, which has a similar number of parameters as the doc2query model, achieves 0.18 BLEU.

row (3) shows that doc2query is able to approach the effectiveness of non-BERT neural models (at the time the work was published) solely with document expansion.



		MS MARCO Passage			
		Development		Test	Latency
Method		MRR@10	Recall@1k	MRR@10	(ms/query)
(1a)	BM25	0.184	0.853	0.186	55
(1b)	w/ doc2query–base [Nogueira et al., 2019b]	0.218	0.891	0.215	61
(1c)	w/ doc2query–T5 [Nogueira and Lin, 2019]	0.277	0.947	0.272	64
(2a)	BM25 + RM3	0.156	0.861	-	-
(2b)	w/ doc2query–base	0.194	0.892	-	-
(2c)	w/ doc2query–T5	0.214	0.946	-	-
(3)	Best non-BERT [Hofstätter et al., 2019]	0.290	-	0.277	-
(4)	BM25 + monoBERT$_{Large}$ [Nogueira et al., 2019a]	0.372	0.853	0.365	3,500

Table 31: The effectiveness of doc2query on the MS MARCO passage ranking test collection.

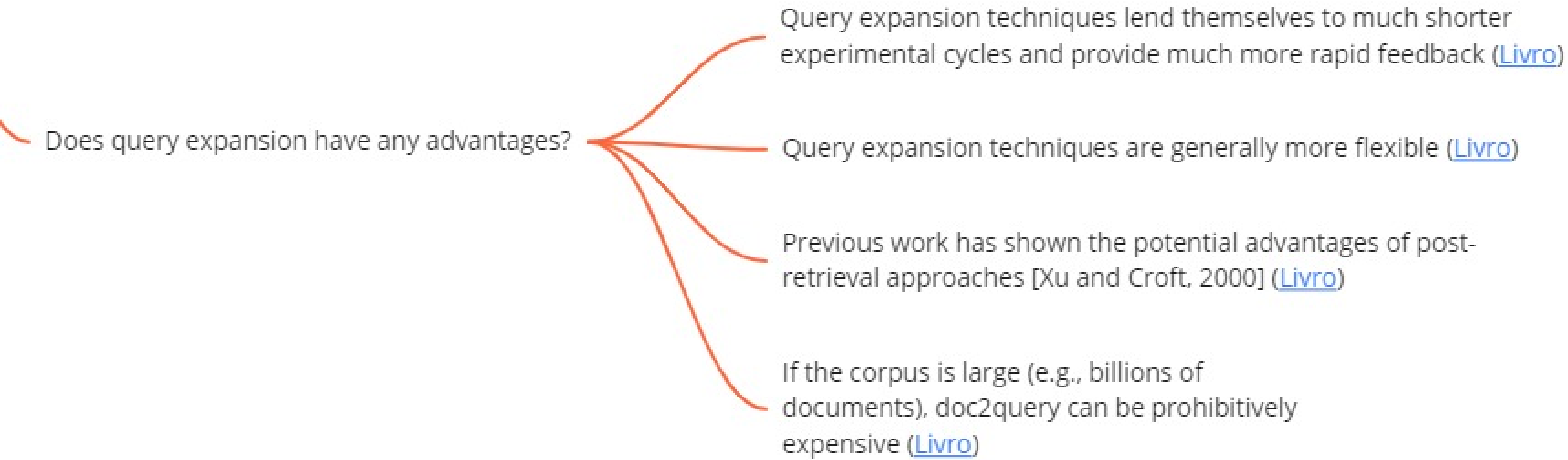


Basic doubts that may arise

What are the advantages of doc2query over query expansion?

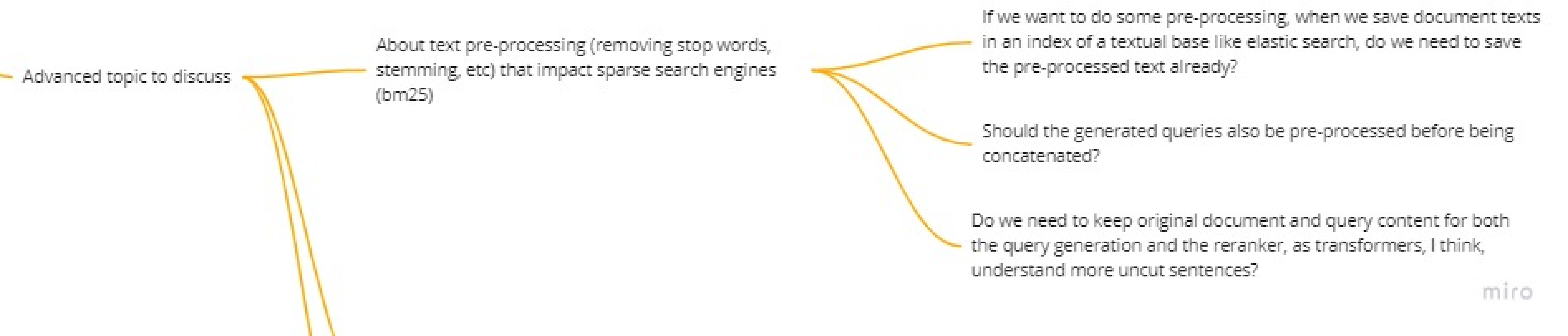
Shorter response time for searches as expansion is done before indexing

"Documents are typically much longer than queries, and thus offer more context for a model to choose appropriate expansion terms." (Livro)



Does query expansion have any advantages?

- Query expansion techniques lend themselves to much shorter experimental cycles and provide much more rapid feedback (Livro)
- Query expansion techniques are generally more flexible (Livro)
- Previous work has shown the potential advantages of post-retrieval approaches [Xu and Croft, 2000] (Livro)
- If the corpus is large (e.g., billions of documents), doc2query can be prohibitively expensive (Livro)



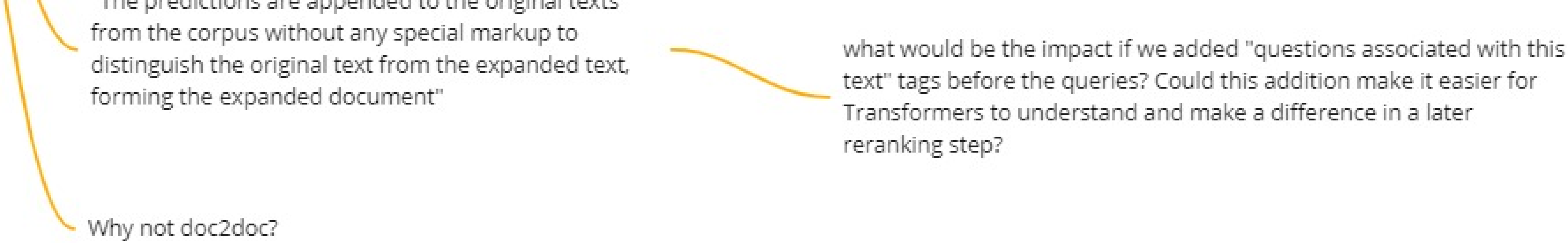
Advanced topic to discuss

About text pre-processing (removing stop words, stemming, etc) that impact sparse search engines (bm25)

If we want to do some pre-processing, when we save document texts in an index of a textual base like elastic search, do we need to save the pre-processed text already?

Should the generated queries also be pre-processed before being concatenated?

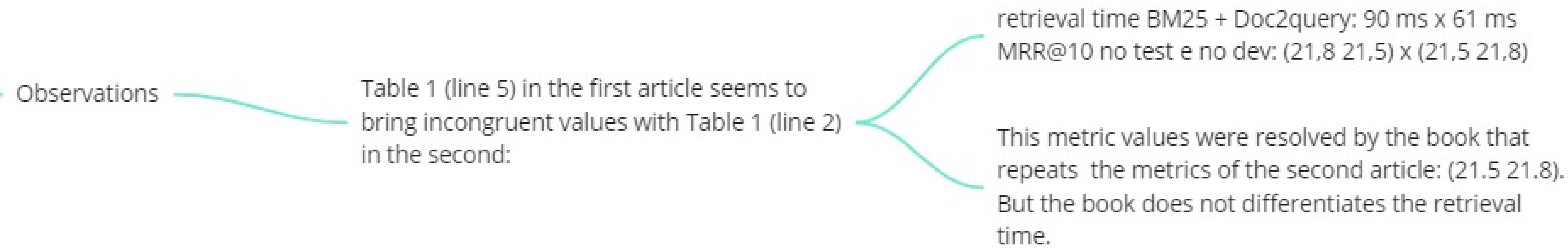
Do we need to keep original document and query content for both the query generation and the reranker, as transformers, I think, understand more uncut sentences?



"The predictions are appended to the original texts from the corpus without any special markup to distinguish the original text from the expanded text, forming the expanded document"

what would be the impact if we added "questions associated with this text" tags before the queries? Could this addition make it easier for Transformers to understand and make a difference in a later reranking step?

Why not doc2doc?



Observations

Table 1 (line 5) in the first article seems to bring incongruent values with Table 1 (line 2) in the second:

retrieval time BM25 + Doc2query: 90 ms x 61 ms
MRR@10 no test e no dev: (21,8 21,5) x (21,5 21,8)

This metric values were resolved by the book that repeats the metrics of the second article: (21.5 21.8). But the book does not differentiates the retrieval time.