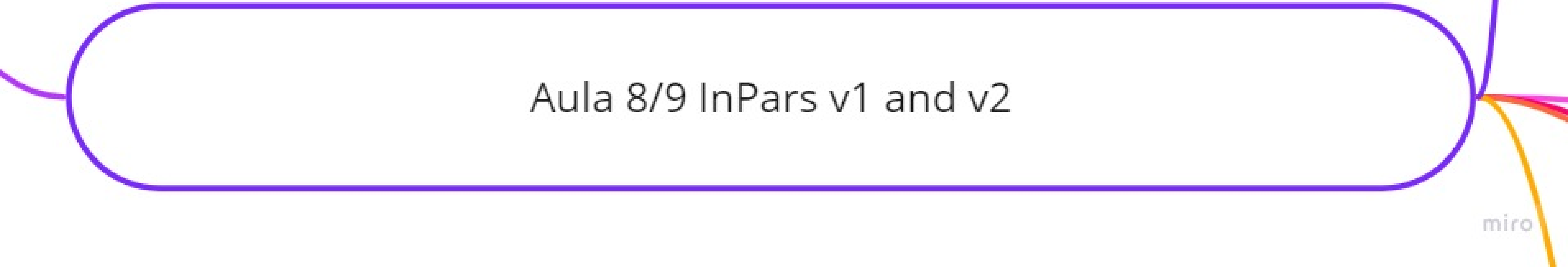
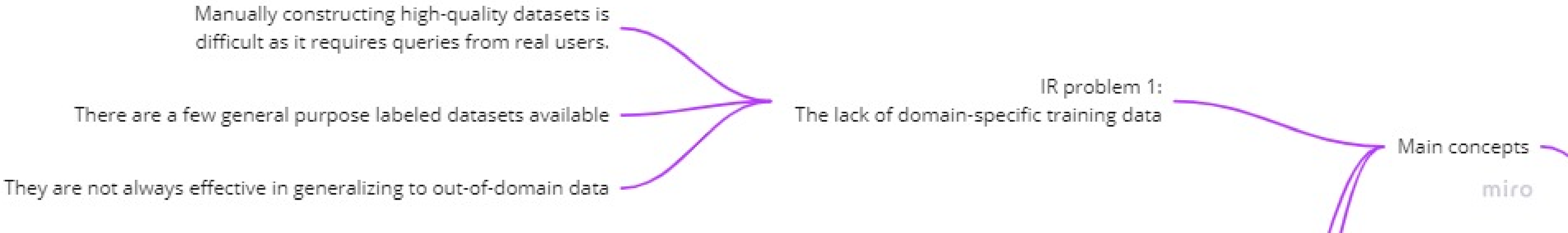
Article Presentation
IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas
Student: Marcus Vinícius Borela de Castro



Aula 8/9 InPars v1 and v2



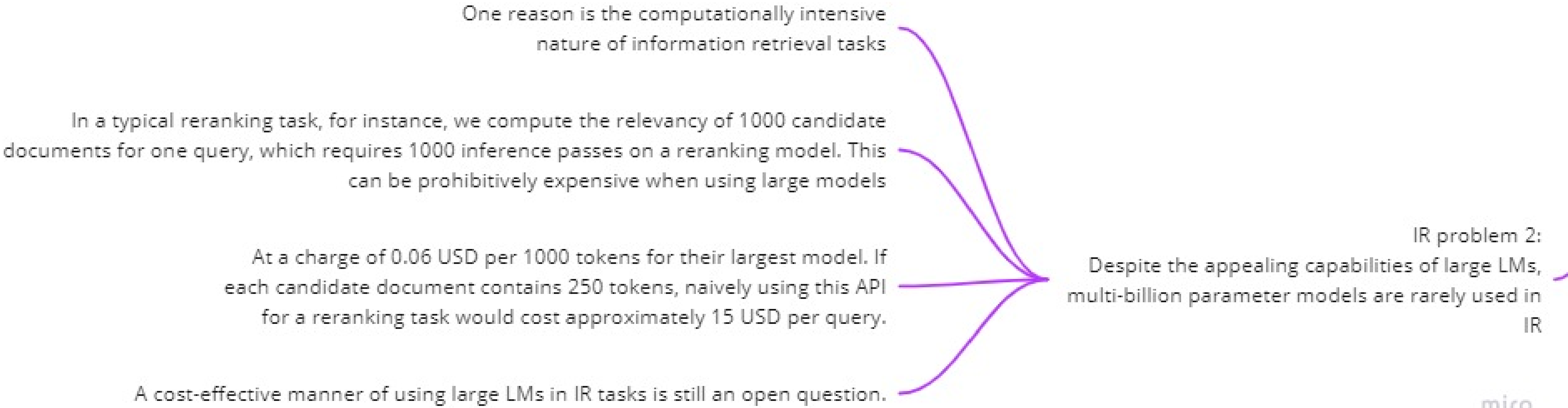
Main concepts

IR problem 1:
The lack of domain-specific training data

Manually constructing high-quality datasets is difficult as it requires queries from real users.

There are a few general purpose labeled datasets available

They are not always effective in generalizing to out-of-domain data



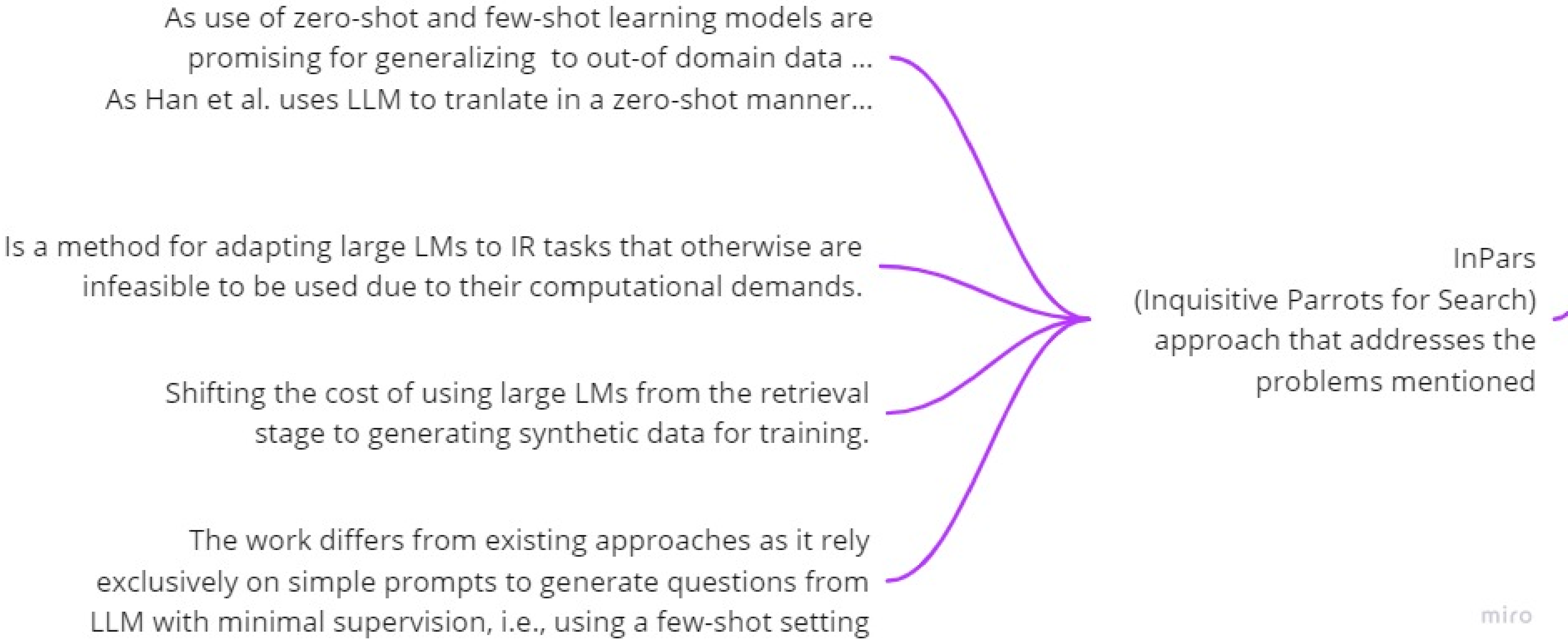
IR problem 2:
Despite the appealing capabilities of large LMs, multi-billion parameter models are rarely used in IR

One reason is the computationally intensive nature of information retrieval tasks

In a typical reranking task, for instance, we compute the relevancy of 1000 candidate documents for one query, which requires 1000 inference passes on a reranking model. This can be prohibitively expensive when using large models

At a charge of 0.06 USD per 1000 tokens for their largest model. If each candidate document contains 250 tokens, naively using this API for a reranking task would cost approximately 15 USD per query.

A cost-effective manner of using large LMs in IR tasks is still an open question.



InPars
(Inquisitive Parrots for Search)
approach that addresses the
problems mentioned

As use of zero-shot and few-shot learning models are promising for generalizing to out-of domain data ...
As Han et al. uses LLM to tranlate in a zero-shot manner...

Is a method for adapting large LMs to IR tasks that otherwise are infeasible to be used due to their computational demands.

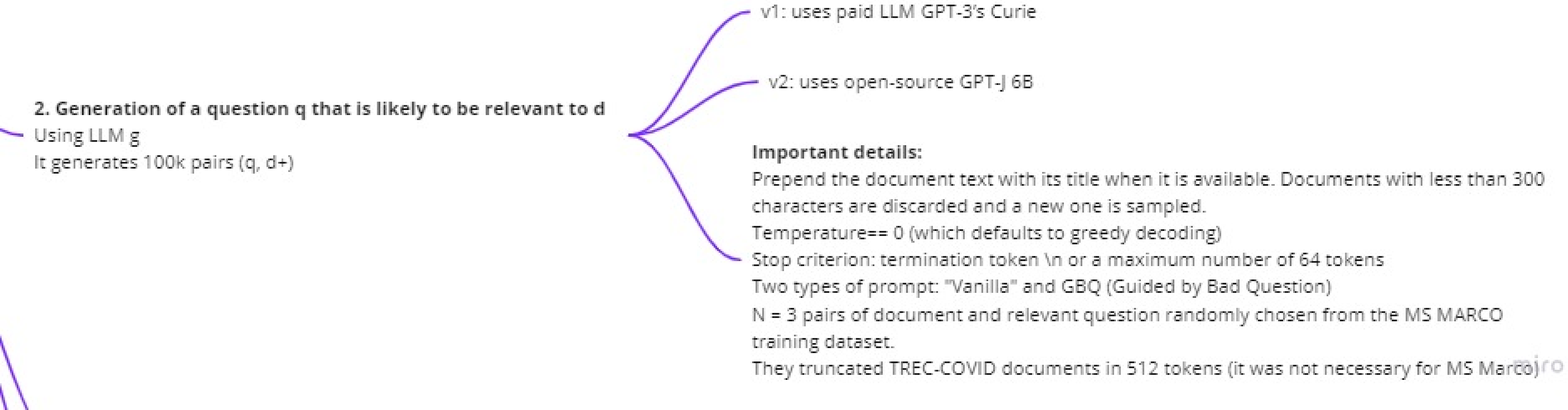
Shifting the cost of using large LMs from the retrieval stage to generating synthetic data for training.

The work differs from existing approaches as it rely exclusively on simple prompts to generate questions from LLM with minimal supervision, i.e., using a few-shot setting

Process
Code on https://github.com/zetaalphavector/inpars.

1. Create a prompt t||d
The concatenation of a prefix t and a document d.
t consists of N pairs of questions and their relevant documents, i.e., t= {(q1, d1), ..., (qN, dN)}
InPairs tried N=3, a few-shot learning approach.

Important details:
The prefix t is always the same regardless of the input document d, i.e., we can potentially generate millions of synthetic training examples using only N manually annotated examples.

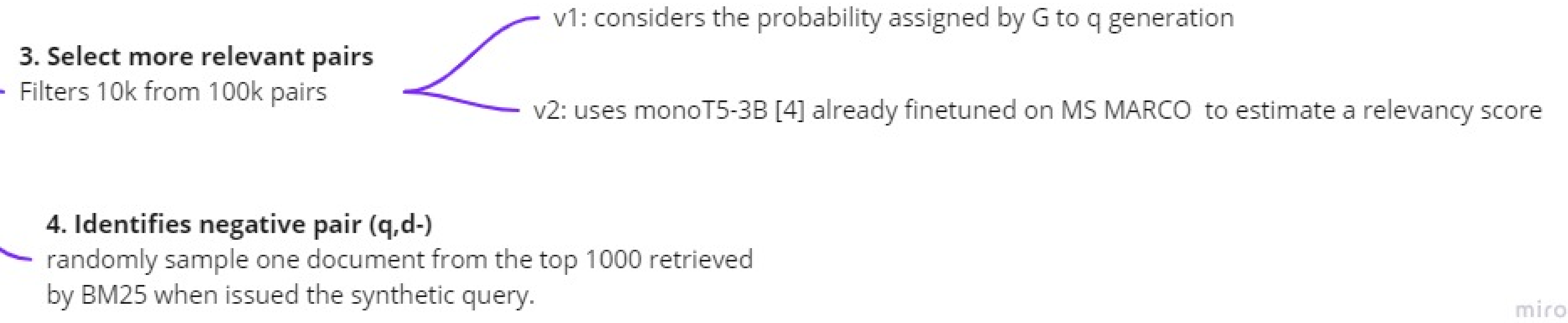


2. Generation of a question q that is likely to be relevant to d
Using LLM g
It generates 100k pairs (q, d+)

v1: uses paid LLM GPT-3's Curie

v2: uses open-source GPT-J 6B

Important details:
Prepend the document text with its title when it is available. Documents with less than 300 characters are discarded and a new one is sampled.
Temperature== 0 (which defaults to greedy decoding)
Stop criterion: termination token \n or a maximum number of 64 tokens
Two types of prompt: "Vanilla" and GBQ (Guided by Bad Question)
N = 3 pairs of document and relevant question randomly chosen from the MS MARCO training dataset.
They truncated TREC-COVID documents in 512 tokens (it was not necessary for MS Marco)

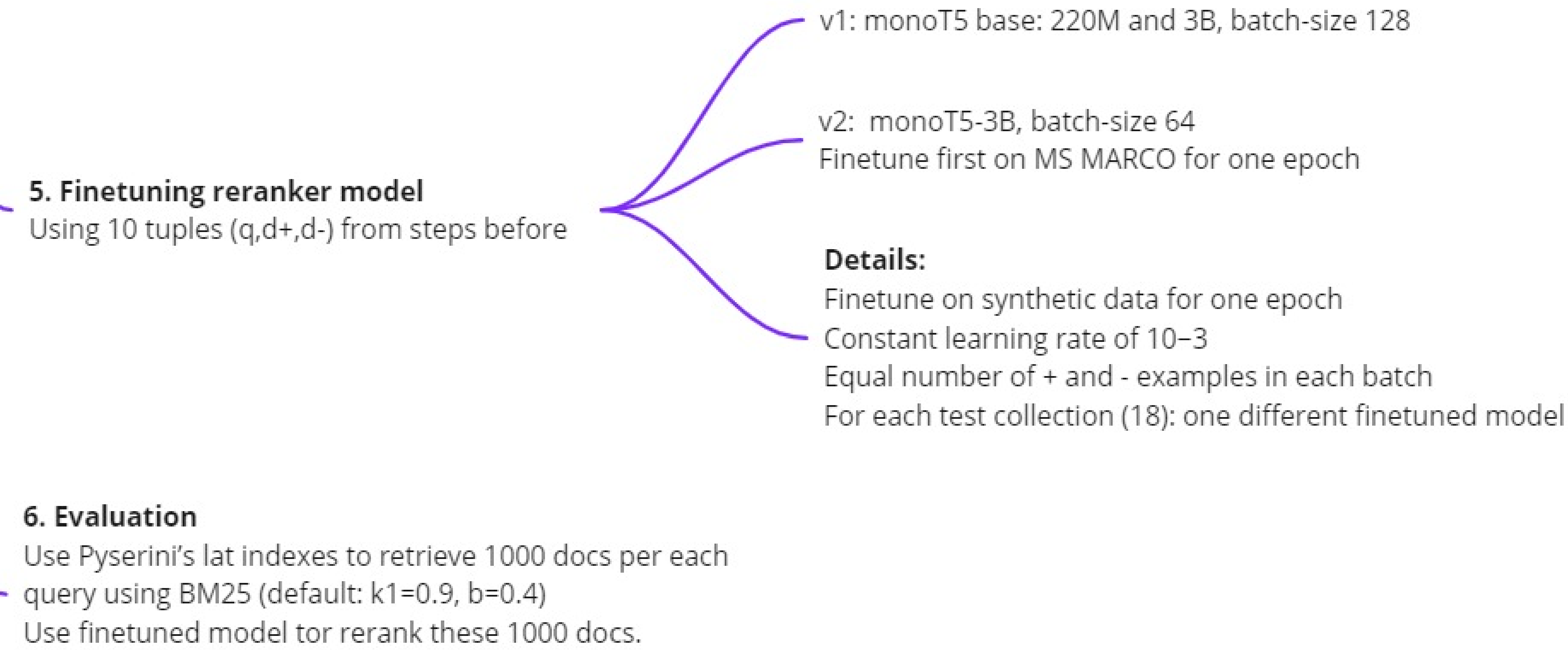


3. Select more relevant pairs
Filters 10k from 100k pairs

v1: considers the probability assigned by G to q generation

v2: uses monoT5-3B [4] already finetuned on MS MARCO to estimate a relevancy score

4. Identifies negative pair (q,d-)
randomly sample one document from the top 1000 retrieved
by BM25 when issued the synthetic query.



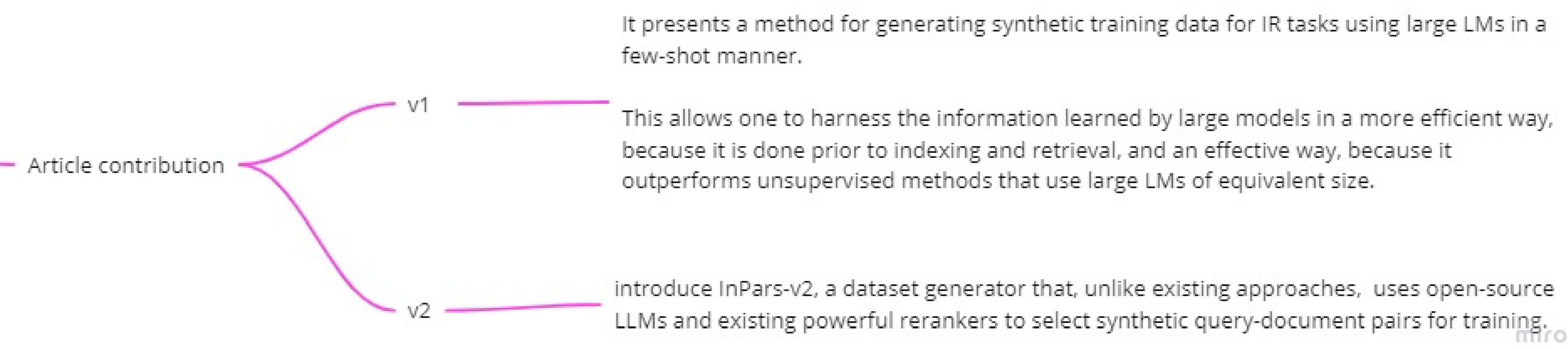
5. Finetuning reranker model
Using 10 tuples (q,d+,d-) from steps before

v1: monoT5 base: 220M and 3B, batch-size 128

v2: monoT5-3B, batch-size 64
Finetune first on MS MARCO for one epoch

Details:
Finetune on synthetic data for one epoch
Constant learning rate of 10−3
Equal number of + and - examples in each batch
For each test collection (18): one different finetuned model

6. Evaluation
Use Pyserini's lat indexes to retrieve 1000 docs per each query using BM25 (default: k1=0.9, b=0.4)
Use finetuned model tor rerank these 1000 docs.



Article contribution

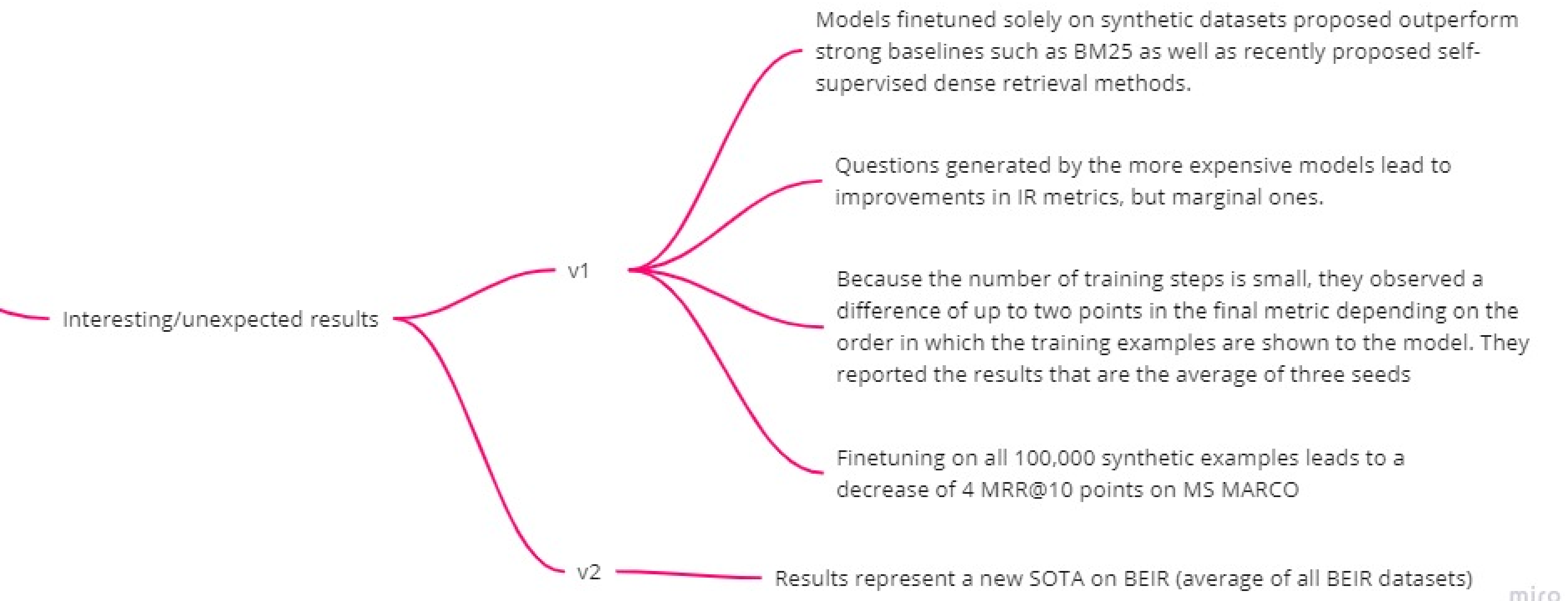
v1

It presents a method for generating synthetic training data for IR tasks using large LMs in a few-shot manner.

This allows one to harness the information learned by large models in a more efficient way, because it is done prior to indexing and retrieval, and an effective way, because it outperforms unsupervised methods that use large LMs of equivalent size.

v2

introduce InPars-v2, a dataset generator that, unlike existing approaches, uses open-source LLMs and existing powerful rerankers to select synthetic query-document pairs for training.



Interesting/unexpected results

v1

Models finetuned solely on synthetic datasets proposed outperform strong baselines such as BM25 as well as recently proposed self-supervised dense retrieval methods.

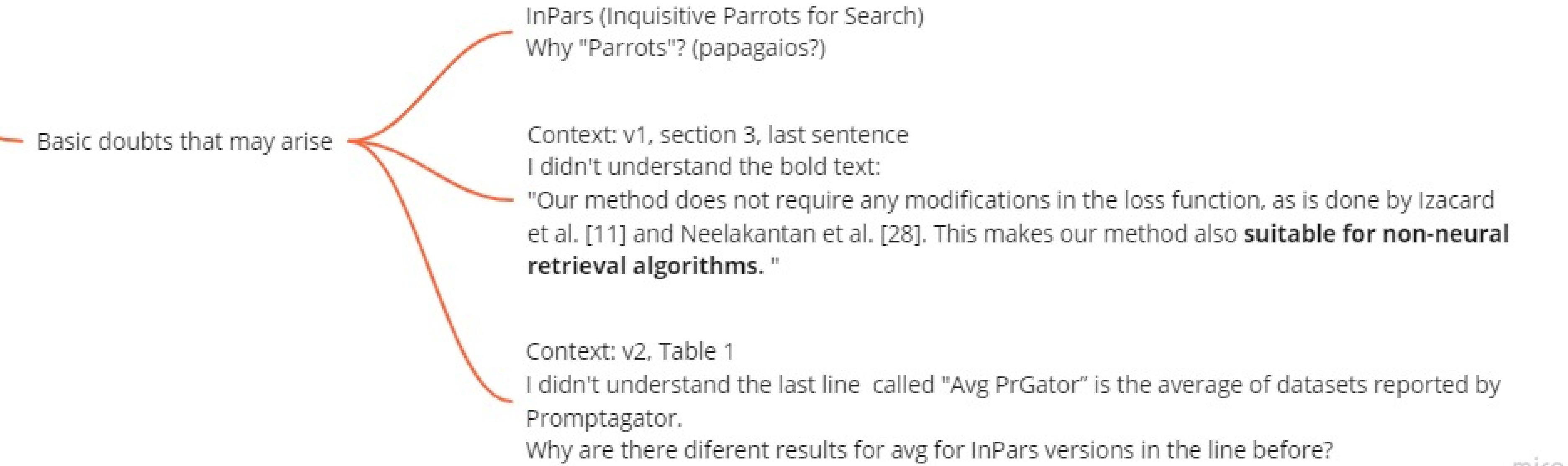
Questions generated by the more expensive models lead to improvements in IR metrics, but marginal ones.

Because the number of training steps is small, they observed a difference of up to two points in the final metric depending on the order in which the training examples are shown to the model. They reported the results that are the average of three seeds

Finetuning on all 100,000 synthetic examples leads to a decrease of 4 MRR@10 points on MS MARCO

v2

Results represent a new SOTA on BEIR (average of all BEIR datasets)

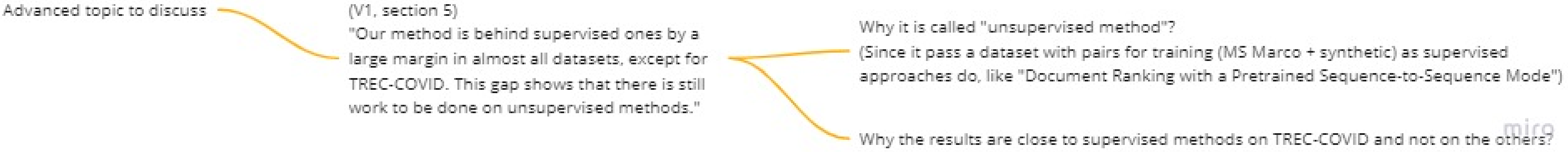


Basic doubts that may arise

InPars (Inquisitive Parrots for Search)
Why "Parrots"? (papagaios?)

Context: v1, section 3, last sentence
I didn't understand the bold text:
"Our method does not require any modifications in the loss function, as is done by Izacard et al. [11] and Neelakantan et al. [28]. This makes our method also **suitable for non-neural retrieval algorithms.** "

Context: v2, Table 1
I didn't understand the last line called "Avg PrGator" is the average of datasets reported by Promptagator.
Why are there diferent results for avg for InPars versions in the line before?



Advanced topic to discuss

(V1, section 5)
"Our method is behind supervised ones by a large margin in almost all datasets, except for TREC-COVID. This gap shows that there is still work to be done on unsupervised methods."

Why it is called "unsupervised method"?
(Since it pass a dataset with pairs for training (MS Marco + synthetic) as supervised approaches do, like "Document Ranking with a Pretrained Sequence-to-Sequence Mode")

Why the results are close to supervised methods on TREC-COVID and not on the others?