

Evaluation and comparison of multilingual and Czech-specific embedding models

NPRG071 Company Project Report

Author: Bc. Roman Tomchik

Faculty: Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University

Study programme: Computer Science - Language Technologies and Computational Linguistics

University supervisor: Mgr. Martin Popel, Ph.D.

Company consultant: Ing. Martin Labsky, Ph.D.

Company: MAMA AI

Abstract

This project implements a configurable evaluation pipeline for Czech and multilingual document retrieval. The pipeline builds on an open-source evaluation framework and extends it with tasks based on custom datasets and an alternate execution path for obtaining embeddings. Sixteen embedding models are evaluated on English passages, their Czech translations, and Czech news articles paired with synthetically generated questions. Five smaller models are additionally evaluated on a broader set of multilingual retrieval tasks. A lexical baseline provides a non-neural comparison. Three cross-encoder rerankers are integrated to reorder the top 100 retrieved news articles, and the combined retrieval and reranking pipelines are evaluated end to end. The results show that model rankings primarily depend on the dataset and that reranking reduces differences between retrievers, allowing to choose a smaller model at the expense of the overall speed.

All the work related to the project can be found on the faculty GitLab repository:

<https://gitlab.mff.cuni.cz/tomchikr/CZ-MTEB>

1. Introduction

This report describes a company project carried out jointly with MAMA AI [1]. The motivation for this project is the pursuit of a best performing embedding model, as well as an attempt to implement a customizable evaluation pipeline, which can be extended further with additional datasets, while maintaining access to existing selection of open-source data.

Text embedding models represent queries and documents as vectors so that relevant documents can be retrieved by vector similarity. For a company developing RAG [2] systems, selecting a model requires evidence on Czech retrieval, not only a model's performance on English or a multilingual aggregate. This project develops an evaluation pipeline and compares model behaviour across translated search passages and native Czech news articles.

1.1. Background and Related Work

Massive Text Embedding Benchmark (MTEB) provides a common framework for evaluating embeddings across tasks and languages [3], while its multilingual extension MMTEB [4] focuses specifically on evaluating models across a broader range of multilingual tasks and languages. This project uses the MTEB Python library [5] as an evaluation engine which comes with open-source datasets hosted via HuggingFace [6], and can be further extended with additional tasks based on the custom datasets and defined by the user in the specific format. This also allows for publishing of the tasks, as well as publishing of scores on already existing tasks.

MS MARCO contains questions derived from real search queries and associated passages [7]. Here, the English queries and their Czech translations provide a passage-retrieval comparison.

The Czech Text Document Corpus (CTDC) contains Czech news articles annotated with topic labels and was originally used for document classification [8]. As part of this project, the corpus was adapted for retrieval benchmarking by generating Czech questions from selected articles and treating each source article as the designated relevant document. These synthetic questions and source-document relevance labels are additional contributions of this project.

The project also drew motivation from the e-INFRA CZ study of embedding-based search over CERIT-SC documentation [9]. This study uses document-conditioned synthetic questions for its evaluation. This present project adds a larger model comparison, two distinct retrieval datasets, a lexical baseline, and a reranking experiment. Its results concern retrieval quality, no generated-answer accuracy is measured.

2. Construction of Datasets

2.1. Datasets

Table 1. Summary of evaluated datasets and model coverage.

Evaluation	Corpus size	Scored queries	Completed model coverage
MS MARCO, Czech query and passage	1,008,985 passages	53,621	16 retrievers
MS MARCO, English query and passage	1,008,985 passages	55,578	16 retrievers
CTDC, Czech query and article	14,690 articles	14,887	16 retrievers
CTDC top-100 reranking	100 articles per query	14,887	16 retrievers x 3 rerankers (48 total)
MTEB multilingual benchmark, retrieval subset	Task-dependent	Task-dependent	5 retrievers

2.2. MS MARCO Construction

The Czech translations of the originally English dataset were created by MAMA AI using a large language model (LLM). This generated dataset is not publicly available and was provided by the company consultant. For each query, there are 10 passages, the ones chosen for this specific query are labeled as 1, the others are labeled as 0. Some queries lacked any passages ranked as positive, such queries only contributed their associated documents to the retrieval corpus and were not used for metric evaluation. Retrieval is performed over this entire corpus for each individual query.

2.3. CTDC Construction

As was mentioned in the Section 1.1, this dataset lacks any questions/queries which can be used for the measurement of the retrieval performance. It was decided to adapt this dataset to a retrieval task by creating a synthetic Q&A dataset, where an LLM will be supplied with a single document from the

corpus and tasked with generating a question to this article. Optionally, an answer can be generated, which might be used for the evaluation of a RAG system end-to-end as a future work.

The synthetic question set was produced using the DeepEval framework's `Synthesizer` component [10] with an Azure-hosted [11] GPT 5.2 model [12], provided by MAMA AI. DeepEval orchestrates the generation of evaluation examples from supplied contexts. In this project, the contexts are existing CTDC articles, the questions and optional answers are generated by continuously querying the Azure service in batches and storing the final dataset in the JSON file.

The designed generation script first constructs a pool of suitable articles. It filters articles shorter than 100 whitespace-delimited words, removes repeated boilerplate news articles, and samples approximately in proportion to the primary article categories in that pool.

Each selected article is then supplied to DeepEval as a separate context. The configured scenario, retrieval task, and style instructions guide the LLM to generate self-contained Czech questions answerable from that article, up to a configured maximum number of questions per article. The questions, contexts, identifiers, and optional answers are all stored in the results JSON file. The retrieval loader uses each question as a query and its source article as the positive relevance label.

The resulting dataset is publicly available as part of the project repository, under `data/synthetic/ctdc_synthetic.json`.

The full DeepEval `StylingConfig` can be found in Appendix A.

2.4. MMTEB Retrieval Datasets

Additionally, we used the retrieval tasks from MMTEB, the multilingual extension of MTEB. Unlike the two handcrafted tasks, these experiments used the benchmark's existing datasets. The goal was to compare the resulting scores with those on the MTEB leaderboard (Section 6) to see whether the results are reproducible.

The selection combines document search with question-answer retrieval and reasoning tasks reformulated as retrieval. Some tasks are English-only, while others evaluate multiple languages or cross-lingual retrieval, where the query and the retrieved text use different languages. The table lists the languages used in these experiments.

Table 2. MMTEB retrieval tasks and evaluated languages.

Dataset/task	Retrieval problem	Languages evaluated
AILAStatutes	Retrieve legal statutes relevant to a described situation.	English
ArguAna	Retrieve a counterargument to a given argument.	English
BelebeleRetrieval	Retrieve passages for questions adapted from multilingual reading comprehension.	115 languages in 122 language/script variants, including Czech and Slovak; complete language list below.
CovidRetrieval	Retrieve relevant COVID-19 news articles.	Mandarin Chinese (Simplified script)
HagridRetrieval	Retrieve manually labelled relevant passages for information-seeking queries.	English
LEMBPasskeyRetrieval	Retrieve long documents containing a requested passkey.	English
LegalBenchCorporateLobbying	Retrieve relevant bills from their titles and summaries in a corporate-lobbying setting.	English
MIRACLRetrievalHardNegatives	Retrieve relevant documents among difficult non-relevant candidates selected by other retrievers.	Arabic, Bengali, Chinese (Simplified), English, Finnish, French, German, Hindi, Indonesian, Japanese, Korean, Persian, Russian, Spanish, Swahili, Telugu, Thai, Yoruba

MLQARetrieval	Retrieve answer-bearing passages for multilingual question-answering queries.	Arabic, Chinese (Simplified), English, German, Hindi, Spanish, Vietnamese
SCIDOCS	Retrieve related scientific papers.	English
SpartQA	Retrieve answers to spatial-reasoning questions.	English
StackOverflowQA	Retrieve responses to programming questions, including responses containing code.	English
StatcanDialogueDatasetRetrieval	Retrieve statistical tables from conversational requests.	English, French
TRECCOVID	Retrieve scientific articles relevant to COVID-19 research questions.	English
TempReasonL1	Retrieve answers to temporal-reasoning questions.	English
TwitterHjerneRetrieval	Retrieve answers to questions posted with the Danish Twitterhjerne hashtag.	Danish
WikipediaRetrievalMultilingual	Retrieve Wikipedia passages using synthetically generated queries.	Bengali, Bulgarian, Czech, Danish, Dutch, English, Finnish, German, Hindi, Italian, Norwegian, Persian, Portuguese, Romanian, Serbian, Swedish
WinoGrande	Retrieve answers to commonsense pronoun-resolution questions.	English

BelebeleRetrieval is the most linguistically diverse task in this selection. Its 115 evaluated languages are:

- Arabic varieties: Egyptian Arabic, Levantine Arabic, Mesopotamian Arabic, Modern Standard Arabic, Moroccan Arabic, and Najdi Arabic.
- Other languages, A-L: Afrikaans, Albanian (Tosk), Amharic, Armenian, Assamese, Azerbaijani (North), Bambara, Basque, Bengali, Bulgarian, Burmese, Catalan, Cebuano, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, Fulfulde (Nigerian), Georgian, German, Greek, Guarani, Gujarati, Haitian Creole, Hausa, Hebrew, Hindi, Hungarian, Icelandic, Igbo, Ilocano, Indonesian, Italian, Japanese, Javanese, Jingpho, Kabuverdianu, Kannada, Kazakh, Khmer, Kinyarwanda, Korean, Kurdish (Central), Kyrgyz, Lao, Latvian (Standard), Lingala, Lithuanian, Luganda, and Luo.
- Other languages, M-Z: Macedonian, Malagasy (Plateau), Malay (Standard), Malayalam, Maltese, Maori, Marathi, Mongolian (Halh), Nepali, Northern Sotho, Norwegian (Bokmal), Nyanja, Odia, Oromo (West Central), Pashto (Southern), Persian (Iranian), Polish, Portuguese, Punjabi, Romanian, Russian, Serbian, Shan, Shona, Sindhi, Sinhala, Slovak, Slovenian, Somali, Southern Sotho, Spanish, Sundanese, Swahili, Swati, Swedish, Tagalog, Tajik, Tamil, Telugu, Thai, Tibetan, Tigrinya, Tsonga, Tswana, Turkish, Ukrainian, Urdu, Uzbek (Northern), Vietnamese, Waray, Wolof, Xhosa, Yoruba, and Zulu.

The 122 variants include both the native-script and Latin-script forms of Modern Standard Arabic, Bengali, Hindi, Nepali, Sinhala, and Urdu, and both Simplified and Traditional Chinese. Belebele was evaluated on same-language query-passage pairs, cross-lingual pairs involving English, and selected cross-script pairs. MLQA likewise includes both same-language and cross-language retrieval across its seven languages.

Czech therefore appears in two of the multilingual tasks: BelebeleRetrieval includes Czech-to-Czech and Czech-English retrieval in both directions, while WikipediaRetrievalMultilingual includes a Czech subset.

3. Methods and Implementation

3.1. Models and Retrieval

Five smaller models were initially selected to run the entire MTEB multilingual retrieval selection of 18 tasks: bge-m3, embeddinggemma-300m, multilingual-e5-large, multilingual-e5-small, and harrier-0.6b (Section 6). These models were subsequently retained for evaluation on the MS MARCO and CTDC tasks, which are based on the datasets described in Section 2. The comparison was expanded with larger models selected from the top of the MTEB leaderboard, restricted to models that had their weights available on HuggingFace.

The complete model identifiers appear in Table 4. Model selection covers several model families and sizes. The model registry reports approximately 560 million parameters for multilingual-e5-large and approximately 27 billion for harrier-27b, illustrating the range of model sizes.

The pipeline loads models through MTEB and delegates encoding and retrieval to its model wrappers. The collected registry metadata for the 16 registered retrievers specify cosine similarity. Cosine similarity is the dot product of two vectors divided by the product of their Euclidean norms.

3.1.1. Query-side Instructions

An instruction tells an embedding model what kind of relationship to retrieve, for example an answer to a question rather than a text on a similar topic. It is part of the encoder input, not a request to generate an answer. The model still returns an embedding, which is compared with document embeddings.

For CTDC, we set the task's query instruction to `Given a question in Czech, retrieve the news article that answers it`. For our MS MARCO tasks, we reuse the native MSMARCO instruction: `Given a web search query, retrieve relevant passages that answer the query`. These descriptions express the intended retrieval objective and are shared across questions within each task. The instructions are written in English, while the query itself remains in its original language.

MTEB passes task metadata and the query/document role to the model wrapper, which applies the model's expected input format. For example, Qwen3-Embedding uses the template `Instruct: {instruction}\nQuery:{query}` for queries. With our CTDC instruction, an illustrative input is:

```
Instruct: Given a question in Czech, retrieve the news article that
answers it
```

```
Query:Kdo vyhrál prezidentské volby v roce 2018?
```

Not every model uses a natural-language task instruction. The distinction follows the model's training and input conventions, not simply its size. Qwen3's instruction-aware wrapper adds the task

description to queries but not to documents. The multilingual-e5-small and multilingual-e5-large wrappers instead use the fixed role prefixes `query:` and `passage:`, without inserting the custom task description. bge-m3 does not require either a task instruction or those role prefixes. We preserve the model-specific conventions.

Native MTEB tasks use the same task-metadata mechanism. Among the tasks evaluated in this project, ArguAna supplies `Given a claim, find documents that refute the claim`, while SCIDOCS supplies `Given a scientific paper title, retrieve paper abstracts that are cited by the given paper`. Instruction-aware wrappers use these descriptions, and models with fixed prefixes or no instruction retain their own input conventions.

3.1.2. Input Lengths

CTDC uses whole news articles rather than pre-cut passages, so long inputs can make encoding expensive in both GPU memory and computation. We introduced token caps to keep these requirements manageable, particularly for larger models. The trade-off is that information outside the retained portion of an article cannot contribute to its embedding.

Setting `max_seq_length = 8192` in the TOML configuration means an upper bound of 8,192 input tokens, not words or characters. After loading the model, the factory sets the model or wrapper's limit to `min(existing_limit, configured_limit)`: a loaded limit of 32,768 becomes 8,192, while a limit of 512 remains 512. The encoder's tokenizer truncates inputs that exceed the effective limit before embedding them. Below are used CTDC limits:

Table 3. Token limits for different models on CTDC.

Token limit	Models
8,192 tokens	bge-m3; both Qwen3 models; both F2LLM models; Octen; jina-v5-text-small; both Harrier models
4,096 tokens	llama-embed-nemotron-8b
2,048 tokens	embeddinggemma-300m
512 tokens	both multilingual-e5 models; KaLM-12B; geevec-lite; Seznam

3.2. Metrics

MRR@K (Mean Reciprocal Rank at K) awards placing the first relevant document early. A single query contributes the reciprocal of the rank of the first relevant passage. If the first relevant passage is not in the first K passages, the contribution is 0. The mean is taken over scored queries.

nDCG@K (Normalized Discounted Cumulative Gain at K) measures ranking quality by penalizing relevant documents that appear lower in the result list, normalizing the score against an ideal ordering.

MAP@K (Mean Average Precision at K) evaluates precision across all relevant ranks up to K, averaged over all levels, penalizing errors earlier in the ranked list.

Recall@K evaluates the proportion of all designated relevant passages for a query that appear within the top K retrieved results, averaged across all scored queries. In RAG applications, this serves as a key metric because the exact rank order of retrieved passages within the context window of the RAG model matters less than ensuring that the important information is successfully captured.

With exactly one positive per question, MAP@K and MRR@K coincide, and Recall@K reduces to a hit rate.

For first-stage retrieval, MTEB uses the model wrappers to encode queries and documents and retrieve documents by similarity score. Its evaluator compares the retrieved document identifiers with the relevance judgements, computes the query-level metrics, and averages them within each evaluated split and subset. Both custom tasks designate MRR@10 as their `main_score`.

The output files separate evaluation scores from individual predictions. The bge-m3 CTDC run provides an example:

- `metrics.json`: the project runner serialises MTEB's returned results, including per-task metric
- `ctdc_synthetic_retrieval_predictions.json`: when prediction saving is enabled, MTEB writes per-query document identifiers and scores.

3.3. Lexical Baseline

BM25 ranks documents by weighted overlap with the query, without computing neural embeddings [13]. We included it as a reference method on the same CTDC questions, articles, and source-document labels used for the embedding models. Since the questions were generated from the articles themselves, shared wording could make retrieval possible through lexical matching. The baseline tests how competitive this approach is.

Queries and documents are lowercased and split into Unicode word tokens; one-character tokens are discarded and the remaining tokens are shortened to their first five characters. This simple prefix heuristic allows some Czech inflected forms to match. The implementation uses BM25 with $k_1=1.5$, $b=0.75$, smoothed inverse document frequency, and each distinct query term counted once.

3.4. Reranking

The rerankers are cross-encoders: each processes a query and one candidate document together and produces a relevance score for that pair. These scores are then used to rearrange the relative ordering of the documents in their relevance ranking for that particular query. This potentially allows to pull more relevant passages into the context window of the RAG model, whereas before it would not fit, and the context window would be instead filled with irrelevant information. Unlike the first-stage embedding models, which encode the two texts separately and compare their vectors, a cross-encoder can use interactions between the query and document tokens when assessing relevance. This requires additional model computation for each query-document pair, which motivates applying it to a shortlist rather than the entire corpus.

The second stage independently scores each query with the first-stage retriever's top 100 documents. The evaluated rerankers are jinaai/jina-reranker-v2-base-multilingual, infgrad/Prism-Qwen3.5-Reranker-0.8B, and mixedbread-ai/mxbai-rerank-large-v2.

Our runner calls the retrieval task's `convert_to_reranking` method with the saved first-stage prediction directory and `top_k=100`. MTEB evaluates the cross-encoder on those query-document pairs through the same `mteb.evaluate` entry point, computing retrieval metrics against the original relevance labels. This reuses the candidate lists without rerunning the embedding model; the runner saves reranking results separately from the first-stage results.

Under the single-source relevance definition, candidate Recall@100 upper-bounds the best possible final MRR@10, since a source absent from the candidate set cannot contribute a positive reciprocal rank.

3.5. Software Architecture

The implementation supports Hugging Face models through MTEB and includes an Ollama embedding adapter. MTEB does not natively support Ollama, so the adapter was added at the request of MAMA AI. It was tested with embeddinggemma-300m on the full MTEB retrieval subset, but was not pursued for the reported evaluation. Running larger models on the local machine was not possible, and running an Ollama server on the cluster was not considered. The Azure language-model backend is used for synthetic data generation.

The processing flow is: TOML configuration, dataset loading, MTEB task construction, model loading, retrieval or reranking, with results stored as JSON files. The entry point is `run.py`. Custom dataset loaders expose queries, documents, and relevance judgements, which are then routed through retrieval tasks written in MTEB format. The runner registers those tasks, invokes `mteb.evaluate`, saves model metrics and benchmark aggregates, and can retain predictions for later reranking and analysis. Slurm and PBS launch scripts support cluster execution.

3.6. UFAL AIC Infrastructure and Execution

The retrieval and reranking experiments used the UFAL AIC cluster at the Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University. GPU inference jobs were submitted through the Slurm scheduler. Each submission selected a per-model TOML configuration, which determined the model, enabled datasets, batch size, and inference settings. CPU-only work, including the debugging, ran in Slurm CPU allocations.

GPU resources were selected according to the model's memory requirements. Large models used automatic device placement (`device_map="auto"`) to distribute weights across multiple allocated GPUs when needed. This enabled evaluation of models that would not fit on a single card.

4. Retrieval Results

The expanded version of this table can be found in results/README.md of the repository. The scores are stored in results/mteb.

Table 4. Retrieval metrics. Rows are ordered by CTDC MRR@10.

Model	MS MARCO CZ MRR@10	MS MARCO EN MRR@10	CTDC MRR@10	CTDC Recall@100
BAAI/bge-m3	0.4386	0.4946	0.7863	0.9874
microsoft/harrier-oss-v1-27b	0.4429	0.5353	0.7829	0.9923
Qwen/Qwen3-Embedding-8B	0.4624	0.5389	0.7816	0.9900
tencent/KaLM-Embedding-Gemma3-12B-2511	0.4260	0.4685	0.7808	0.9887
nvidia/llama-embed-nemotron-8b	0.4938	0.5699	0.7789	0.9880
Qwen/Qwen3-Embedding-4B	0.4453	0.5315	0.7735	0.9869
microsoft/harrier-oss-v1-0.6b	0.3994	0.5281	0.7716	0.9866
Octen/Octen-Embedding-8B	0.4464	0.5191	0.7608	0.9887
intfloat/multilingual-e5-large	0.4672	0.5447	0.7573	0.9874
jinaai/jina-embeddings-v5-text-small	0.4406	0.5277	0.7431	0.9794
intfloat/multilingual-e5-small	0.4054	0.5174	0.7242	0.9749
codefuse-ai/F2LLM-v2-8B	0.4567	0.5465	0.6994	0.9780

codefuse-ai/F2LLM-v2-14B	0.4585	0.5495	0.6970	0.9799
google/embeddinggemma-300m	0.3680	0.4981	0.6892	0.9619
geevec-ai/geevec-embeddings-1.0-lite	0.3138	0.5111	0.5900	0.9220
Seznam/simcse-retromae-sm-all-cs	0.1465	0.1110	0.4689	0.8968
BM25, five-character prefixes	Not evaluated	Not evaluated	0.7871	0.9884

4.1. Czech and English MS MARCO

Nemotron has the highest observed MRR@10 in both languages, followed on Czech by multilingual-e5-large and Qwen3-8B. All 15 non-Seznam models score higher in English than in Czech. Seznam scores higher in Czech, but has the lowest absolute score in both languages, which can be explained by the fact that this model is relatively old and not comparable to recent multilingual options.

The results do not support a simple rule that a larger model always retrieves better. For example, multilingual-e5-large scores 0.4672 on Czech MS MARCO, above F2LLM-14B at 0.4585 and harrier-27b at 0.4429.

4.2. Lexical Baseline

BM25 reaches MRR@10 of 0.7871, Recall@1 of 0.6968, and Recall@100 of 0.9884. The result shows that a neural model is not necessary to achieve a competitive aggregate score on this synthetically-generated CTDC task.

Table 5. CTDC questions grouped by the BM25 rank of their designated source.

BM25 source rank	Questions	Share
1	10,373	69.68%
2-10	3,631	24.39%
11-100	710	4.77%
Greater than 100	173	1.16%

BM25 recovers the source within ten results for 94.07% of questions. Neural models also recover some sources that BM25 misses. On the 883 questions where BM25 places the source below rank ten, $MRR@10$ is 0.3448 for harrier-27b, 0.3433 for KaLM, and 0.3414 for bge-m3.

4.3. Rank Transfer Between Datasets

Across all 16 retrievers, the Spearman correlation between Czech and English MS MARCO is 0.847. Between Czech MS MARCO and CTDC it is only 0.406. bge-m3 moves from tenth on Czech MS MARCO to first on CTDC, whereas F2LLM-14B moves from fourth to thirteenth. KaLM moves from eleventh to fourth, and multilingual-e5-large from second to ninth. These reversals make a single averaged Czech score difficult to interpret for model selection.

The datasets differ simultaneously in corpus size, document length, query origin, language provenance, and relevance annotation. CTDC's lexical baseline suggests that lexical matching is important there, but the current data do not isolate it as the cause of the rank swaps.

4.4. Retrieval Failures and Complementarity

In cases when the source is absent from the top ten, 397 questions (2.67%) are failed by all five leading CTDC retrievers, while 1,723 (11.57%) are failed by at least one of them. The intersection-to-union ratio is 0.230. Across all 16 retrievers, 234 questions (1.57%) are failed by every model. The figures indicate potential improvement when models are combined, but no fusion or ensemble was evaluated in these experiments. Combining candidate sets is a reasonable future experiment.

5. Reranking Results

MAMA AI indicated that Recall@20 around 0.98 would be a good practical target for a production RAG system. We therefore tested whether reranking the top 100 candidates could bring the source article into the final 20 results often enough to reach that level. On CTDC, with one labelled source per question, Recall@20 of 0.98 means that the source appears among the first 20 results for 98% of questions.

5.1. Comparing the Same First-Stage Models

We select bge-m3 and Seznam as the highest- and lowest-scoring first-stage embedding models by CTDC MRR@10, respectively, and follow those same two models through all three rerankers.

Table 6. Before and after reranking for the fixed model pair.

Retriever	Reranker	MRR@10 before / after	MRR@10 gain	Recall@20 before / after
BAAI/bge-m3	Jina	0.7863 / 0.8637	+0.0774	0.95795 / 0.98139
BAAI/bge-m3	Prism	0.7863 / 0.8455	+0.0591	0.95795 / 0.97924
BAAI/bge-m3	mxbai	0.7863 / 0.8114	+0.0250	0.95795 / 0.97078
Seznam/simcse-re tromae-small-cs	Jina	0.4689 / 0.7994	+0.3305	0.76825 / 0.89407
Seznam/simcse-re tromae-small-cs	Prism	0.4689 / 0.7809	+0.3120	0.76825 / 0.89306
Seznam/simcse-re tromae-small-cs	mxbai	0.4689 / 0.7615	+0.2926	0.76825 / 0.89125

bge-m3 reaches the Recall@20 target with Jina. Seznam makes a much larger MRR gain, but remains below the target with every reranker: its source article is present in the top-100 candidates

for only 89.682% of questions. Reranking cannot recover the missing sources, so even perfect ordering of those candidates could not bring its Recall@20 to 0.98.

5.2. Reaching the Recall@20 Target

No first-stage embedding model reaches Recall@20 of 0.98 on CTDC. After reranking, nine models reach it with Jina, three with Prism, and none with mxbai. The table shows each model's starting Recall@20 and the value after each reranker. Recall@100 is unchanged by reranking and provides the candidate-recovery ceiling for all three runs.

Table 7. Recall@20 before and after reranking.

Retriever	Recall@ 20 before	After Jina	After Prism	After mxbai	Recall@100 ceiling
BAAI/bge-m3	0.95795	0.98139	0.97924	0.97078	0.98737
microsoft/harrier-oss-v1-27b	0.96527	0.98603	0.98227	0.97642	0.99228
Qwen/Qwen3-Embedding-8B	0.95795	0.98388	0.98039	0.97508	0.98999
tencent/KaLM-Embedding-Ge mma3-12B-2511	0.95949	0.98287	0.97998	0.97340	0.98865
nvidia/llama-embed-nemotron- 8b	0.95808	0.98166	0.97844	0.97212	0.98798
Qwen/Qwen3-Embedding-4B	0.95499	0.98106	0.97810	0.97165	0.98690
microsoft/harrier-oss-v1-0.6b	0.95264	0.98079	0.97763	0.97065	0.98657
Octen/Octen-Embedding-8B	0.95399	0.98294	0.97945	0.97387	0.98865
intfloat/multilingual-e5-large	0.95405	0.98173	0.98012	0.97367	0.98744
jinaai/jina-embeddings-v5-text- small	0.93780	0.97374	0.97085	0.96588	0.97938
intfloat/multilingual-e5-small	0.93088	0.97011	0.96910	0.96326	0.97488

codefuse-ai/F2LLM-v2-8B	0.93101	0.97333	0.97058	0.96567	0.97803
codefuse-ai/F2LLM-v2-14B	0.93377	0.97501	0.97273	0.96735	0.97992
google/embeddinggemma-300m	0.90636	0.95755	0.95553	0.95023	0.96191
geevec-ai/geevec-embeddings-1.0-lite	0.83482	0.91852	0.91758	0.91341	0.92195
Seznam/simcse-retromae-small-cs	0.76825	0.89407	0.89306	0.89125	0.89682

For the seven retrievers whose initial Recall@100 is below 0.98, no reranker restricted to these same candidates can meet the strict target. Improving their first-stage candidate recovery, for example by retrieving more documents or combining retrievers, would be needed to raise that ceiling. The performance and cost of such changes were not evaluated here.

6. Multilingual Benchmark Evaluation

Five smaller embedding models were also tested on 18 retrieval tasks from Section 2.4 to check whether the scores reported on the MTEB leaderboard could be reproduced. Table 6 shows the mean score across these tasks, calculated using MTEB's `Mean(Task)` aggregation, which takes the main metric for each task in the selection and returns their mean.

The acquired scores did not fully match the leaderboard. The most likely reason is due to differences in benchmark versions, which can change the underlying dataset versions. Since it was not the purpose of this work to isolate what specifically caused the mismatch, it was not investigated further.

Table 8. MTEB retrieval-selection aggregate scores for 18 retrieval tasks.

Model	Mean(Task)
microsoft/harrier-oss-v1-0.6b	0.7117
google/embeddinggemma-300m	0.6058
BAAI/bge-m3	0.5638
intfloat/multilingual-e5-large	0.5626
intfloat/multilingual-e5-small	0.5088

harrier-oss-v1-0.6b leads this subset but ranks thirteenth on Czech MS MARCO and seventh on CTDC. This is further evidence against using a multilingual aggregate alone to select a Czech retriever.

7. Conclusion and Future Work

The project delivers a configurable retrieval evaluation pipeline, capable of using both native MTEB and Ollama for acquiring the embeddings, synthetic Czech query-document dataset, results for 16 retrievers on two custom tasks, and 48 CTDC reranking experiments, as well as a lexical baseline BM25. The practical value of the project is a development of a testing pipeline with a large customization potential that can be reused and updated to meet further development needs.

The conclusions from the performed experiments are:

- Model rankings depend primarily on the dataset, which supports the idea of an aggregate evaluation on multiple diverse retrieval datasets.
- On the CTDC synthetic benchmark, BM25 is competitive in comparison to the best document-embedding based approaches, which supports its introduction as a baseline in all future experiments.
- Reranking substantially improves the final performance, which allows avoiding using larger embedding models.
- Recall@K for some larger K can be a useful measure of the upper bound of the performance of the system.

For MAMA AI, the main recommendation is a selection procedure, as one universal winning model cannot be inferred from the experiments. The procedure is to evaluate candidate systems on representative queries and documents, include BM25 as a baseline, and decide whether the production system will use a reranker, and if yes, on what count of documents will reranking be performed. The default metric for a one-stage system can be, for example, Recall@20, while for a two-stage system Recall@K can be used as an upper bound on the performance of the entire system, where K is the amount of documents being reranked.

The follow-up work is, in order of importance:

1. Create a large human-reviewed company-domain evaluation set. Label multiple relevant documents and inspect ambiguous questions.
2. Evaluate BM25-plus-reranker system and an ensemble of multiple retrievers, as the current failure overlap motivates these experiments.
3. Run the experiments on the Czech datasets native to the MTEB benchmark.

References

- [1] MAMA AI. *Enterprise AI Solutions for Smart Business Automation*. Accessed 17 September 2026. <https://themama.ai/>
- [2] Lewis, P., Perez, E., Piktus, A., et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. NeurIPS 2020. <https://arxiv.org/abs/2005.11401>
- [3] Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. *MTEB: Massive Text Embedding Benchmark*. arXiv:2210.07316, 2022, revised 2023. <https://arxiv.org/abs/2210.07316>
- [4] Enevoldsen, K., Chung, I., Kerboua, I., et al. *MMTEB: Massive Multilingual Text Embedding Benchmark*. arXiv:2502.13595, 2025. <https://arxiv.org/abs/2502.13595>
- [5] MTEB Maintainers. *MTEB: Massive Text Embedding Benchmark Python Library*. GitHub repository. Accessed 17 September 2026. <https://github.com/embeddings-benchmark/mteb>
- [6] Hugging Face. *Hugging Face Hub documentation*. Accessed 17 September 2026. <https://huggingface.co/docs/hub/en/index>
- [7] Bajaj, P., et al. *MS MARCO: A Human Generated Machine Reading Comprehension Dataset*. arXiv:1611.09268, revised version, 2018. <https://arxiv.org/abs/1611.09268>
- [8] Král, P., and Lenc, L. *Czech Text Document Corpus v 2.0*. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 4345-4348, 2018. <http://ctdc.kiv.zcu.cz/lrec18.pdf>
- [9] Blaško, Š. *AI docs search / Embedders*. e-INFRA CZ Blog, 27 June 2025. Accessed 17 September 2026. <https://blog.e-infra.cz/blog/embedders/>
- [10] Confident AI. *DeepEval: Golden Synthesizer documentation*. Accessed 17 September 2026. <https://deepeval.com/docs/synthesizer-introduction>
- [11] Microsoft. *Foundry Models sold by Azure*. Microsoft Learn, Azure OpenAI documentation. Accessed 17 September 2026. <https://learn.microsoft.com/en-us/azure/foundry/foundry-models/concepts/models-sold-directly-by-azure>
- [12] OpenAI. *GPT-5.2*. Model documentation. Accessed 17 September 2026. <https://developers.openai.com/api/docs/models/gpt-5.2>
- [13] Robertson, S., and Zaragoza, H. *The Probabilistic Relevance Framework: BM25 and Beyond*. Foundations and Trends in Information Retrieval, 2009. <https://doi.org/10.1561/1500000019>

Appendix A. DeepEval Styling Configuration

```
from deepeval.synthesizer.config import StylingConfig

STYLING_CONFIG = StylingConfig(
    scenario=(
        "A Czech speaker asking a question about a specific event, fact or figure "
        "reported in one article of the ČTK news agency archive."
    ),
    task=(
        "Retrieve the single most relevant Czech news article from the archive and "
        "answer the user's question from it."
    ),
    input_format=(
        "A single self-contained question in Czech, between 6 and 20 words, ending "
        "with a question mark. It must ask about exactly one fact and be answerable in "
        "one or two sentences. It must never chain several sub-questions together with "
        "commas, colons, or connectives such as 'a kdy', 'a jak', 'a kolik'. It must "
        "paraphrase the article in the user's own words instead of quoting the headline "
        "or any sentence verbatim. It must never contain instructions such as 'shrň', "
        "'ověř', 'vytvoř tabulku' or 'představ si', must never mention 'text', "
        "'kontext', 'článek' or 'dokument', and must not use markdown. It must name "
        "enough entities and details that only this one article answers it."
    ),
    expected_output_format=(
        "A short factual answer in Czech, at most two sentences, stating only the fact "
        "the question asks for, with no extra commentary. Plain text only, no markdown "
        "formatting, no bullet lists and no tables."
    ),
)
```