Adapting Multilingual Semantic Search to the Chemical Process Industry Domain

Master Thesis

Anna Bringmann
anna.bringmann@stud.uni-goettingen.de
Georg-August University Göttingen
Germany

ABSTRACT

Textual information retrieval systems based on multilingual dense retrievers hold the potential to break down language barriers and close the lexical gap. However, large-scale labelled data is needed to train dense retrievers and they are sensitive to domain shifts. For most specialised domains, the availability of labelled data is scarce and collecting it is resource-intensive. Thus, automated processes for domain adaptation in low-resource setups are required. This thesis experimentally explores possibilities for performing domain adaptation of multilingual semantic search in a zero-shot setting utilising unlabelled Germand and English documents from the chemical process industry and a small labelled test set for model evaluation. We compare the influence of four variables on domain adaptation in a multilingual setup. The first variable is the query-generation method. We compare an extractive approach sampling keyword queries from the corpus based on the underlying document language model. An inventive approach that uses a pre-trained multilingual t5-based generative model to generate queries and a prompt-based approach utilising a GPT-4oinstance to create queries. The second variable we explore is the influence of using knowledge distillation to pseudo-label the generated dataset to account for the possibly poor quality of generated queries. The third variable we explore is the influence of model size on domain adaptation. We fine-tune a small LLaMA-based embedding model and a larger XLM-RoBERTa model with the generated datasets, analysing the effect of their size on their domain-adaptation capacities. The fourth and last variable is the influence of the dataset language (German or English). Our domain-adaptation approaches yielded no systematic improvements for the English data. For the German data, however, multiple combinations of query generation approaches, pseudo-labelling, and model architecture lead to improved retrieval performance compared to the best baseline model. Our best-performing model improves the retrieval of German target domain data by 3.58

KEYWORDS

multilingual, semantic search, query generation, domain adaptation, low resource, zero-shot, dense retrieval

1 INTRODUCTION

Text information retrieval (IR) describes acquiring the relevant textual information from all available textual information, given a user-entered natural language query. The query and the text are a word sequence divided into several tokens from a specific language [131].

Traditionally, textual IR was performed using lexical methods, such as probabilistic relevance frameworks like BM25 [94, 95]. However, these methods suffer from the lexical gap since they do not include semantic information and, thus, cannot recognise synonyms or distinguish ambiguous words. Nowadays, natural language processing (NLP) is widely applied to textual IR tasks to close the gap. The most common models are *dense retrievers*, and their invention resulted in the concept of semantic search. Dense retrieval models map the user query and the available textual information to a shared dense vector space. The generated representations include semantic information and are closer to one another in the shared vector space if they are semantically similar. Thus, dense retrievers utilise (approximate) nearest neighbour search to fetch relevant information. However, large-scale labelled data is needed to train dense retrievers, and they are sensitive to domain shifts [107].

Ramponi and Plank [86] define a *domain* as a coherent type of text corpus, i.e. the specific dataset used for training. The type is a variety of latent factors, e.g. topic (chemistry vs. sports), genre (social media vs. news article) or style (formal vs. informal) [86]. For most specialised domains, the availability of labelled data is scarce and collecting it is resource-intensive. Doing that manually for every language and every domain is not feasible [102]. Thus, automated processes for *domain adaptation* in low-resource setups are required and have been subject to extensive research in the past years [91, 102, 131]. Nonetheless, most of these approaches are developed on and tested for monolingual, English, dense retrieval, while the field of multilingual semantic search is rapidly advancing at the same time, holding the promise of breaking down language barriers and expanding information access across diverse linguistic contexts [10, 26, 61, 89, 120, 129].

However, the possibility of combining domain-adaptation techniques with multilingual approaches to semantic search has yet to be explored. Therefore, this project aims to experimentally examine the advantages and disadvantages of different domain adaptation techniques when utilised in a multilingual setup. Our project is, thus, experiment-driven, and our research question, "How to perform domain adaptation for multilingual semantic search in a low-resource setup?" formalises our objective.

We defined our target domain through two small annotated chemical process industry test datasets, one containing German and one containing English query document pairs.

The chemical process industry encompasses various sectors involved in transforming raw materials into valuable products through chemical reactions and processes. This industry is critical in producing chemicals, petrochemicals, pharmaceuticals, and materials like plastics and composites, with applications across diverse fields such as agriculture, manufacturing, and energy. It involves complex, large-scale industrial processes that often require continuous operations. Plants within this industry must operate 24/7, producing significant amounts of data from various sources, including sensors, logs, shift notes, and observations from field operators. The operations rely on seamless collaboration between multiple departments to ensure efficient production and maintain safety standards. Given the intricate nature of chemical processes, the continuous generation of data and the need for uninterrupted operation, information transfer across shifts and departments is a pivotal aspect of the industry. Data gathered from automated systems and human inputs—such as shift logs, process notes, and operational observations—can provide critical insights into the plant's performance. However, this information is often vast and unstructured, making it difficult for operators and decision-makers to extract the most relevant insights, especially when dealing with process upsets or emergencies. Reliable access to this information is crucial, as it directly impacts knowledge transfer, decision-making, and production safety. Semantic search systems can address the challenge of knowledge transfer by enabling more efficient and accurate retrieval of domain-specific information from historical logs. However, labelled datasets from the chemical process industry domain for dense retriever training are rarely available. However, we can use the vast amount of generated data during operations to create labelled datasets.

Thus, apart from the small labelled testset, we were also provided extra unlabelled German and English documents from the chemical process domain. We further describe the datasets in section 3.

In our analysis of how to adapt multilingual dense retrievers to the chemical process domain in a low-resource setup, we experimentally compare four variables: query generation methods, relevance labels' effect, model size, and language.

For the first variable, query-generation methods, we compare an extractive approach sampling keyword queries directly from the unlabelled document corpus vocabulary based on the underlying document language model. Qgen [69], a state-of-the-art inventive query generation approach that uses a pre-trained multilingual t5-based generative model to generate queries and a prompt-based approach utilising a GPT-4o instance to create queries.

For the second variable, the influence of pseudo-relevance labels, we apply the GPL method suggested by Wang et al. [113], using knowledge distillation to pseudo-label the generated datasets, accounting for the possibly poor quality of the generated queries.

For the third variable, model size, we fine-tune two multilingual dense retrievers, differing in size and architecture, on the generated unlabelled and pseudo-labelled datasets — one small LLaMA-embedding model [1] with a BERT base and another larger XLM-RoBERTa-based model (XLM-R) [2]. We further describe the models in section 4.3.

To explore the fourth and last variable, language, we apply all other variables separately to the German and English data. Thus, we generate language-specific datasets and train the two multilingual dense retrievers for each language separately, analysing the effect and interaction of language and the other variables.

In section 5.1, we further explain our experimental setup;

Our overall best-performing model reaches a retrieval performance averaged over Precision@10, Recall@10, F@10, MRR@10, average Precision@10 and ndcg@10[3] of 71.65% on the German test data. That is an improvement of 3.58% compared to the best-performing German baseline model. The best-performing model is a multilingual XLM-R model with 278M model parameters and a dense vector dimension of 768. We fine-tuned it on German positive query document pairs, created by a t5-encoder-decoder model without relevance labels using Multiple Negatives Symmetric Ranking loss.

Our Contributions are:

For the German data, we ...

- ... extend QGen [69] to a multilingual semantic search setup leading to better retrieval performance in a highly specialised domain
- ... verify Ni et al.'s [80] finding that larger models adapt better to new domains in low resource [80] settings.
- ... show that pseudo-relevance labels can improve query results, but query generation methods without any relevance labels already lead to substantial performance boosts
- ... verify Dai et al.s [23] finding that enriching the prompt with knowledge about the nature of the unlabelled documents improves the quality of the generated queries

However, our domain adaptation approaches do not improve the retrieval performance for the English data, likely due to the larger amount of English data used for training multilingual dense retrievers, making the possible retrieval improvement of multilingual English dense retrievers when applied to English data relatively small, even for out-of domain data. In section 5 we explain our results in detail.

The rest of this paper is structured as follows: In section 2, we present related work to give an overview of the research field, exploring the possibilities of adapting multilingual semantic search to specialised domains. Afterwards, in section 3, we explore our unlabelled and test data. In section 4, we introduce our method. In section 5, we introduce and discuss our specific experimental setup and findings for each experiment, summarising the advantages and disadvantages of the analysed domain adaptation techniques and discussing interactions and individual effects of the four analysed variables: query generation, pseudo-labelling, model size and language. Then, we finalise our work by drawing a section 6 and giving an outlook in section 7.

To make this thesis more comprehensive, independent of readers' knowledge levels, we included a short history of performing textual information retrieval in appendix D and recap the architecture of information retrieval systems and dense retrievers in appendix E.

2 RELATED WORK

A growing body of research describes dense information retrieval systems [48, 68, 131] and the topic of adapting dense retrievers to new domains in low-resource settings gained attention within the last years [53, 57, 113, 119].

The following table [11] categorises existing domain-adaptation approaches:

[1] https://huggingface.co/thuan9889/llama_embedding_model_v1

[2] https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1

[3] We introduce all metrics in appendix A

Table 1: Domain-Adaptation Approaches

Adaptation Level	Method	Data Requirements Target Domain
Data	Query Generation [13, 23, 31, 59, 69, 87, 102, 113, 122, 134]	Only Text
	Knowledge Distillation [16, 17, 17, 40, 42, 49, 60, 68, 84, 106, 106, 113]	Text and Queries
Model	Size [126]	Labelled Data
	Capability [80]	Labelled Data
Training	Negative Selection [38, 48]	Positive Query Document Pairs
	Multi-task [3, 27, 75]	Labelled Data
	Domain-Invariant [18, 28, 29, 32, 64, 67, 110, 111]	Labelled Data
	Parameter-Efficient [15, 43, 54, 58, 65, 85, 105]	Labelled Data
Ranking	Integrating Sparse Retrieval [16, 21, 48, 51, 85, 91, 115, 119]	Labelled Data

Since we focus on data-level and model-level approaches to domain adaptation, this section introduces related work on both strategies.

2.1 Data Level Domain-Adaptation

Several studies focus on generating labelled datasets for low-resource domains. While query generation methods use unlabelled documents to generate queries given a document, knowledge distillation approaches automatically establish missing links between queries and documents, forming (pseudo) query-document pairs and generating relevance judgments, making the data more informative for model training. In practice, query generation and knowledge distillation are often combined.

2.1.1 *Query Generation*.

We distinguish two main query generation approaches. Extractive approaches utilise various parts of the unlabelled documents or external knowledge as (pseudo) queries. Inventive approaches use a generative model to create (pseudo) queries.

2.1.1.1 Extractive Approaches

Extractive approaches differ in the part of the unlabelled document or the type of external knowledge from which they extract the synthetic query.

Summary-based methods extract a key segment from the unlabelled document to form the (pseudo) query. For example, the document title [73, 74], random sentences from the first section [15], n-grams [15], or keywords [71].

Proximity-based methods focus on positional proximity within the unlabelled document. One approach, ICT, selects a random sentence from a passage as the query and the remainder as the positive document [54]. Other methods involve selecting random spans, sentences, or paragraphs from the unlabelled document as pseudo queries [30, 70].

Hyperlink-based methods leverage anchor-document relationships to generate query-document pairs, assuming they reflect query-document relevance [69, 130]. Examples include using the first section of a document as a query and hyperlink-connected sentences from another document as positive pairs [15, 123]. Despite its effectiveness, the availability of hyperlinks limits this approach [104].

2.1.1.2 Inventive Approaches

We distinguish inventive query generation approaches by the type of model they use to generate the (pseudo) query.

Prompt-based approaches utilise prompt-based generative LLMs, which can be presented with documents and a prompt to generate a query [9, 22, 23, 96] For example, Dai et al. 2022b utilise FLAN to generate pseudo-queries for model training in a few-shot setting and a zero-shot setting. The authors prompt FLAN to create queries. The prompt includes a description of the search intent (e.g., find a counterargument and answer the query), an unlabelled document and, for the few-shot setting, an annotated query document example. Bonifacio et al. 2022 also generate prompt-based query using gpt-3 as the generative LLM.

Shakeri et al. [101] utilise a pre-trained language model and reframe the task of question-answer generation to machine reading comprehension. Based on that, they train a seq2seq network that generates a question-answer pair given an input text. Reddy et al. [87] adapt this idea and add a selection step to the approach, enabling them to generate better question-answer pairs.

The QGen method [69] uses a query generator trained on general domain data to generate domain queries for the target corpus.

Alberti et al. [5] use a large text corpus to construct question-answer-pairs in three stages: First answer extraction, second question generation and third roundtrip filtering. Lewis et al. [55] extended this approach by adding passage selection and global filtering.

2.1.2 *Knowledge Distillation*.

The general definition of knowledge distillation in machine learning is to transfer knowledge from a more capable model (teacher) to a less capable model (student). For domain adaptation of dense retrievers, we use a teacher to generate (more) precise (pseudo) relevant judgments for the query document pairs. Afterwards, the generated (pseudo) labelled data is used to train the student model [17, 106, 113].

Different models function as teachers in a knowledge distillation setting. For example, pre-trained cross-encoder [40, 42, 84] or enhanced bi-encoder (e.g. ColBERT [49]) [16, 17, 60, 68, 106].

GPL [113] adds knowledge distillation to the QGen method [69] by generating (pseudo) labels from a cross-encoder (the teacher) and uses them as soft labels to train a dense retriever (the student). Combining knowledge distillation with query generation is especially useful since the pseudo-relevance scores can be used to filter out generated queries that are irrelevant to the document.

2.2 Model Level Domain-Adaptation

In addition to tackling the problem of domain-adapting dense retrievers in low-resource scenarios on the data level, we can adapt the models themselves. Zhan et al. [126] showed that cross-encoders are more capable of generalising to out-of-domain test data than bi-encoders and sparse retrievers. However, their high computational costs during inference make their practical application impractical. Ni et al. [80] scaled up the size of a T5 dense-retriever through multi-stage training and reached a better zero-shot retrieval performance on several BEIR [107] datasets. They suggest that bigger models are more capable of adapting to new domains than smaller models.

2.3 Multilingual Dense Retriever

While all of the approaches mentioned above use monolingual, English, dense retriever, the field of multilingual semantic search grows [7, 14, 26, 61, 76, 89, 120, 129], independent from the field of adapting dense retrievers to new domains. Whereas [37] considers multilingual dense retrieval and adapting dense retrievers to domains with specialised language in low-resource setups, they do not investigate the possibility of combining both.

In general, developing multilingual semantic search systems is challenging due to the scarcity of labelled datasets in languages other than English. Notable exceptions include mMARCO [10], a machine-translated multilingual version of MS MARCO with 13 languages, and Mr. TYDI [128], covering 11 languages.

Multilingual dense retrievers are trained on monolingual texts in multiple languages, allowing a single model to generate multilingual text representations. It is possible to fine-tune these models on high-resource languages and apply them to low-resource ones in zero-shot settings. Adding small amounts of target language data to the fine-tuning procedure further improves performance. The essential mechanism of multilingual dense retrieve is that they create a shared multilingual embedding space, mapping different language embeddings to similar feature vectors [36, 46, 52, 78].

LaBSE [26], a recent multilingual model, generates 768-dimensional sentence embeddings and has 471 million parameters. It excels in sentence alignment due to its multilingual pre-training, but its high dimensionality and large parameter count make it costly to adapt for downstream tasks. To address this, LEALLA [77], a distilled encoder model, produces lower-dimensional multilingual sentence embeddings with reduced computational demands while maintaining competitive performance. The authors train LEALLA using LaBSE through knowledge distillation, combining feature and logit distillation.

LASER [6] takes a different approach by using an encoder-decoder LSTM model trained on a translation task, excelling in exact translation tasks but less effective with non-exact sentence similarity. Another approach is mUSE [19], trained on general domain data with a translation ranking task. However, it requires hard negatives for optimal performance, resulting in higher computational costs [35]. Reimers and Gurevych [89] also explore knowledge distillation to extend monolingual models into multilingual ones.

However, studies exploring the possibilities of adapting multilingual semantic search systems to low-resource domains are scarce. Thus, this project offers a new perspective on combining approaches from different research branches.

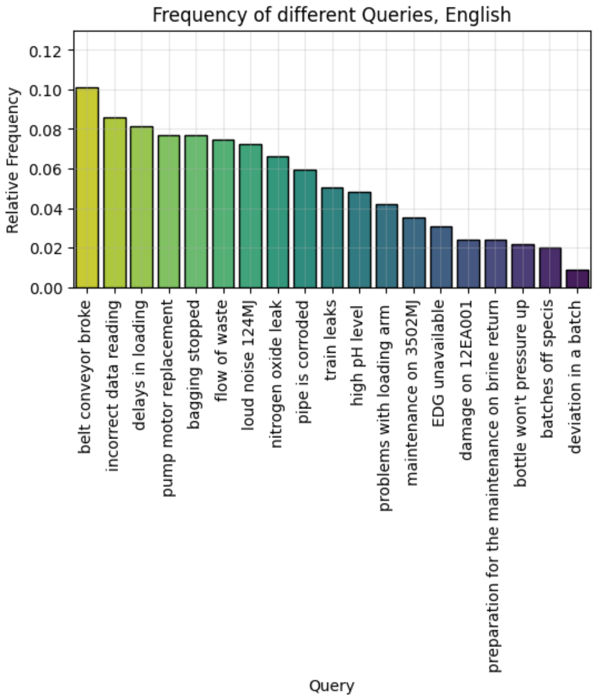

Figure 1: English Queries

3 DATA EXPLORATION

Since our data exploration informs some of our methodological choices, we introduce it in the following sections.

3.1 Labelled Testdata

The test sets contain 19 unique German queries, matching 405 different German documents and another 19 unique English queries, matching 455 different English documents. 11 English queries lead to the same documents. However, only two German queries share the same document as a "response ".

The test data queries have a simple sentence structure:

The most frequent English query accounts for 10Another feature of the two query datasets is that they are not direct translations of each other, making models trained for translation tasks, such as LaBSE [26], less suitable for data processing. The most frequent German query contains only one word[4], while the most frequent English query includes three words and an average number of tokens per query of 3.211. The German queries generally consist of one to two words; the longest consists of five words. Some English queries are complete sentences, but most also consist of two to three words; the longest contains seven words. Therefore, the queries we generate for model training should also consist of one to seven words for the English data and one to five for the German data. Overall, the test queries consist of keywords instead of a complete sentence or a question, making the data very suitable for keyword search and not as ideal for semantic search due to the not very complex queries [131]. However, the query data also contains a

[4] I use the terms word and token interchangeably

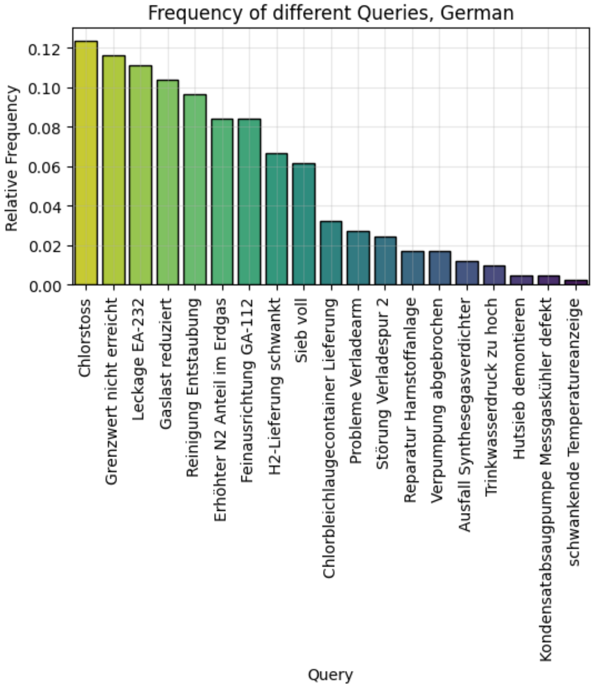

Figure 2: German Queries

difficulty score for each query document pair. This query difficulty is a complexity score, describing how a query matches the relevant document on an ordinal 1-3 scale:

(1) simple, i.e., a query contains some terms directly present in a document,
(2) medium, i.e., a query contains some synonyms to the terms from a document,
(3) hard, i.e., a query matches a document on an abstract level, i.e., if a query contains "problem with a pump", it can match a document that reports about leakage in a pump.

Figure 3 and fig. 4 visualise the query difficulty distributions.

The difficulty of queries also differs between the two languages. 58% of the German queries have a difficulty score of one, while 85% of the English queries have a difficulty of one. Thus, for both languages, one is the most frequent category. However, 21% of the German queries have a difficulty of two and another 20% a difficulty of three. In the English query data, both categories each contain less than 8% of the query data. Thus, English queries nearly always contain words that also occur in the document, making it rather suitable for a search based on lexical similarity. However, that is different for the German query data, even though the simplicity of the queries' structure would suggest otherwise. Since we want our textual information retrieval to be comparable across the two languages, we will focus on dense retrieval instead of a lexical similarity approach, as it should also be appropriate for the English data and the only approach applicable to the German data.

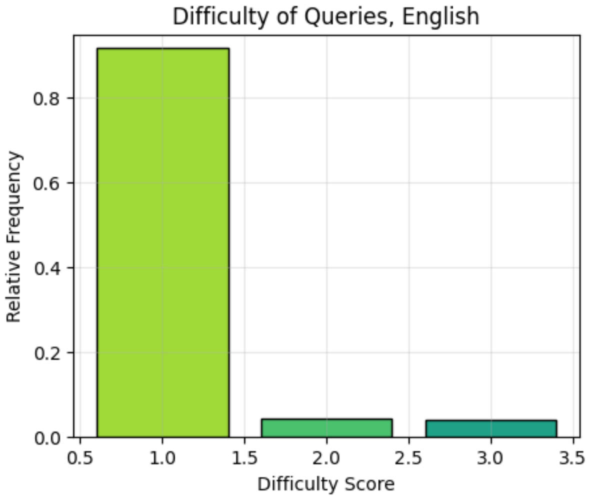

Figure 3: English Queries

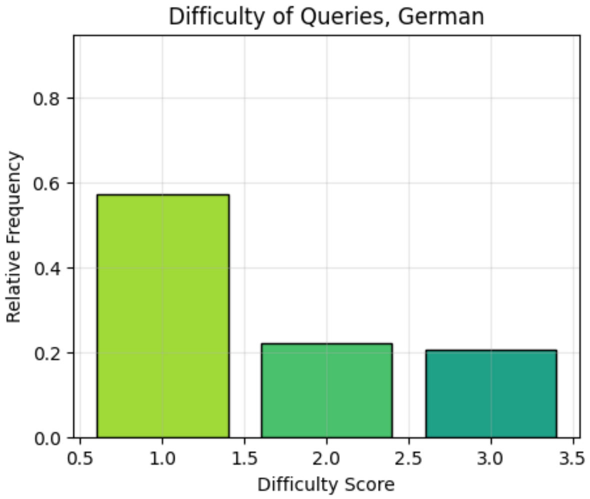

Figure 4: German Queries

3.2 Unlabelled Documents

In section 4 we formalised the document corpus as $D = \{d_1, d_2, ..., d_m\}$, where d_i represents an individual document within the corpus. For our bilingual set-up D is actually further divided into $D = \{D_{DE}, D_{EN}\}$, where:

- $D_{DE} = \{d_1^{DE}, d_2^{DE}, ..., d_m^{DE}\}$ represents the German sub-corpus containing documents written in German, and
- $D_{EN} = \{d_1^{EN}, d_2^{EN}, ..., d_n^{EN}\}$ represents the English sub-corpus containing documents written in English

The two separate sub-corpora D_{DE} and D_{EN} consist of documents that capture human-written information related to the operations, issues, and maintenance activities occurring within chemical and process industry plants[5]. The primary function of these documents is to ensure that critical operational knowledge is preserved across shift changes, minimising the risk of information loss.

More specifically, the documents in each sub-corpora include:

- Operational Logs: Records detailing ongoing work processes, including descriptions of work operations completed, operational statuses, and tasks performed by different personnel during their shifts.
- Repair and Maintenance Reports: Information regarding equipment issues, troubleshooting steps taken, repairs performed, and related technical details.
- Problem and Incident Reports: Descriptions of problems or incidents encountered during plant operations, including observed faults, their potential causes, and any remedial actions.
- Questionnaires and Forms: Occasionally, the documents may include filled-out questionnaires or structured forms, mainly when documenting routine inspections or safety checks.
- Informal Greetings and Announcements: Short, non-operational messages including greetings and well-wishes (e.g., "Happy New Year," "Merry Christmas", "Happy Easter"). These messages serve a social or morale-boosting purpose but are unrelated to the plant's technical or operational content.

The English corpus contains 47,145 unlabelled documents, and the German one contains 79,201 unlabelled documents. However, to eliminate the "number of documents" as a confounder variable, we use 20,000 unlabelled documents for query generation for each language.

4 METHOD

We formalise the textual IR task as:
Given the natural language query q and a collection of m documents $D = d_i{}_{i=1}^{m}$ the IR system returns a list $L = [d_2, d_2, ..., d_n]$ of n relevant documents sorted by the retrieval model's relevance scores. To calculate these relevance scores, sparse information retrieval models perform lexical matching, while dense retrievers perform semantic matching.

Based on this formalisation of textual IR systems, a way of formalising dense retrievers arises:
Dense retrieval models encode the query and the document corpus as dense vectors. Thus, they compute relevance scores through some similarity function between these dense vectors:

$$\text{Rel}(q, d) = f_{sim}(\phi(q), \psi(d)) \tag{1}$$

where $\phi(\cdot) \in \mathbb{R}^l$ and $\psi(\cdot) \in \mathbb{R}^l$ are functions that map the query and the documents into l-dimensional vector space. A deep neural network performs this mapping, and the similarity f_{sim} is measured by, for example, the inner product or the cosine similarity. Dense retrievers, thus, measure the semantic interaction of query and text based on the learned representation of both in latent semantic space. The closer the query and text are to one another in latent space, the more similar they are [131].

As shown in table 1, we need to generate queries through either an extractive or a more complex generative approach to use the unlabelled German and English documents from the chemical process industry for domain adaption.

4.1 Query Generation

Dai et al. [23] analyse that information retrieval tasks can take different forms depending on the search intent and the query distribution Q. Both define how the query and document pair match. For example, question-answering tasks require the information retrieval system to retrieve passages answering the passed question. In contrast, some argument retrieval tasks search for support, and others search for counterarguments to an argument passed as the query. Therefore, we want our generated queries to be as similar to the test queries as possible to fit the real-life search intent and query distribution $Q = \{q_1, q_2 \ldots, q_k\}$. We analyse the test data and the unlabelled documents in section 3.

4.1.1 ***Extractive Query Generaion.***

Since the queries in the original data consist of several keywords instead of entire sentences, we utilise an extractive query generation approach to construct synthetic queries. Since extractive approaches primarily generate queries consisting of keywords and should thus create data similar to the original one.

Sun et al. [104] found that utilising a hyperlink-based approach to construct queries from documents performs best. However, we cannot apply this method since hyperlink information is unavailable in our data.

Another limiting factor is that our documents only contain short passages, while many extractive learning methods rely on rather long documents containing multiple passages. Thus, only perturbation methods or sampling keywords of random n-grams from the unlabelled documents are methods applicable to our data.

4.1.1.1 **Keyword Query Generation**

To generate queries as similar as possible in query distribution to those in our test set, we will apply the keyword method suggested by Ma et al. [71]. Based on the document language model, they predict representative keywords for a given document. This approach assumes that search users have a reasonable intention of what terms appear in the "ideal" document that satisfies their information needs. The generated keywords as the query are thus the pieces of text that represent this "ideal" document.

Using Bayes Theorem, we can formulate this query-generation idea as a probabilistic model:

$$P(d|q) \propto P(q/\Theta_d)P(d) \tag{2}$$

Where:

- $q = q_1, ..., q_m$ is the query
- $d = w_1, ..., w_n$ is a document
- Θ_d the document language model estimated for every document
- $P(d)$ the prior (usually assumed to be uniform and can thus be ignored)

[5] One plant for each sub-corpus

The uniform prior assumption simplifies the formula so that the query likelihood $P(q/\Theta_d)$, which is the query generation probability given the document language model, approximates the relevance of a document to a query $P(d|q)$.

The document language model found best performing for query sampling is the multinomial unigram language model [124]. Each word of the query is generated independently. The formal notation of the query likelihood is thus:

$$P(q|d) = \prod_{i}^{m} P(q_i|\Theta_d) = \prod_{w \in V}^{i} P(w|\Theta_d)^{c(w,q)} \tag{3}$$

Where:

- V is the corpus vocabulary
- $c(w, q)$ is the count of word w in query q

Smoothing techniques are applied to eliminate zero probabilities for unseen words and improve the accuracy of the estimated document language model. For keyword queries, Dirichlet prior smoothing works best [125]. We formalise it as:

$$P(w|d) = \frac{c(w, d) + \mu P(w|C)}{|d| + \mu} \tag{4}$$

Where:

- $c(w, d)$ is the count of word w in document d, the term frequency (tf)
- μ is a smoothing parameter defined as the average number of words in a document for the whole corpus
- $P(w|c)$ is a background (collection) language model based on word counts in the entire document collection, the document frequency (df)
- $|d|$ is the length of the document (total word count)

We use this document language model with prior smoothing to sample a set of words given the input document. Each sampled set of words is a generated pseudo-query, but the word set with the highest likelihood is deemed most "representative" of the document.

We sample a positive integer l from a Poisson distribution as the word set length (the number of keywords to sample from the document) to simulate the varying query length. We formalise the Poisson distribution as:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x}, x = 1, 2, .., \tag{5}$$

Where:

- λ is a hyper-parameter indicating the interval's expectation.

Ma et al. [71] use $\lambda = 3$. The average number of tokens per query in the English test data is 3.211, making $\lambda = 3$ adequate for sampling the query length. However, the average German query length in the test data is 2.263. Thus, we set $\lambda = 2$ for sampling the query length for the German data.

Thus, for each word set, we independently sample l words from the corpus vocabulary $V = \{w_i\}_1^N$ according to the multinomial unigram language model with Dirichlet prior smoothing.

We use to perform the keywords query extraction:

We describe our implementation of algorithm 1 in section 4.4.

A downside of this keyword generation approach is that the query keywords are drawn with replacement, leading to search queries with repeated tokens, which makes them less close to the queries in the test set and to real-life search queries. Even though the generated keyword queries are similar in structure to the test queries, it remains questionable whether using these simple queries for fine-tuning a pre-trained large language model (LLM) is enough to enhance the semantic, domain-specific representation it generates, as the synthetical queries themselves will only contain words which can also be found in the corpus vocabulary, as they are only extracted, not generated. Therefore, at least for the German data, the test set contains more difficult queries than the ones generated by this approach, as shown in fig. 4. Thus, we compare the positive query document pairs generated through the extractive learning approach to two inventive query generation approaches that create more diverse and complex queries using pre-trained large language models.

Algorithm 1: Sampling a Pair of Representative Word Sets

```
Input :Document $d$, Vocabulary $V = \{w_i\}_1^N$, probability
       of word $w_i$ generated by the document language
       model with Dirichlet smoothing $P(w_i|d)$, Query
       likelihood score function $QL(w_i, d)$
//Choose length
$l$ = Sample($X$), $X \sim$ Poisson($\lambda$), $x = 1, 2, 3, ...$
$S_1, S_2 = \emptyset, \emptyset$
//Paired sampling
for $k \leftarrow 1$ to $l$ do
    $S_1 = S_1 \cup$ Sample($V$), $w_i \sim P(w_i|d)$
    $S_2 = S_2 \cup$ Sample($V$), $w_i \sim P(w_i|d)$
end
//Higher likelihood deemed more representative
$S_1$ score = $\prod_i^l QL(w_i, d), w_i \in S_1$
$S_2$ score = $\prod_i^l QL(w_i, d), w_i \in S_2$
if $S_1$ score > $S_2$ score then
    Output: $S_1, d$
else
    Output: $S_2, d$
end
```

4.1.2 *Inventive Query Generation*.

Compared to the previously introduced extractive approach, the inventive generated queries are not extracted from the text passages. Instead, they are created by a pre-trained large language model, given a text passage. Hence, their wording is more diverse, making the generated queries closer to a semantic search use case and real-life search intents. Models trained on this kind of synthetic data reach performances close to fully supervised models [102, 113].

Different inventive query generation methods exist, depending on the large language model used to generate the queries. We use a prompt-based generative model for query generation and adapt QGen, a state-of-the-art approach using a t5-based encoder-decoder for query generation [69], to our multilingual setup. In the following sections, we introduce botch approaches.

4.1.2.1 **Prompt-Based Query Generation**

Dai et al. [23], and Bonifacio et al. [10] showed that prompt-based query generation leads to excellent retrieval performance, with Promptagator even exceeding the performance of GPL evaluated with ndcg@10 on different BEIR datasets. [113]. Therefore, we also apply a prompt-based query generation method.

We formalise the prompt-based query generation problem as:

Following our notation in section 3.2, $D = \{D_{DE}, D_{EN}\}$ is the document corpus

Where:

- $D_{DE} = \{d_1^{DE}, d_2^{DE}, \ldots, d_m^{DE}\}$
- $D_{EN} = \{d_1^{EN}, d_2^{EN}, \ldots, d_n^{EN}\}$

Although we used language-specific prompts, we will simplify the problem by focusing on a monolingual scenario in the following notation. Thus here $D = \{d_1, d_2, \ldots, d_n\}$

Let $\mathcal{P}$ be a set of two different prompts:

$$\mathcal{P} = \{p_1, p_2\} \tag{6}$$

For each document $d_i \in D$, the goal is to generate a set of q search queries using a generative LLM. We perform this generation by applying each prompt $p_j \in \mathcal{P}$ to each document d_i, producing two sets of queries for each document.

We define $G(d_i, p_j)$ as the process that generates q search queries for document d_i using prompt p_j, where G is the generative LLM:

$$G(d_i, p_j) = \{q_1^{d_i,p_j}, q_2^{d_i,p_j}, \ldots, q_k^{d_i,p_j}\} \tag{7}$$

Where:

- $q_k^{d_i,p_j}$ is the k-th query generated for document d_i using prompt p_j.

We set k to three. Thus, three synthetic queries are generated for each document d_i and each prompt p_j.

We define the prompt-based query generation task as generating two sets of queries for each document d_i:

$$\forall d_i \in D, Q_1^{d_i} = G(d_i, p_1) \quad \text{and} \quad Q_2^{d_i} = G(d_i, p_2) \tag{8}$$

Where D is the document corpus, Q_1 and Q_2 are the two sets of generated queries, and p_1 and p_2 are the two prompts used for query generation.

We use algorithm 2 to generate the prompt-based queries:

We had access to a GPT-4o [82] instance and thus used that as our generative LLM G. For a free, open-source version, Dai et al. [23] show that FLAN [116] can replace GPT. For the GPT-4o instance, we set the top_p nucleus sampling to 0.65, the frequency penalty to 0.0005, the presence penalty to 0.0005 and the temperature to 0.65.

4.1.2.2 **QGen**

Generative Pseudo Labelling [113], a state-of-the-art approach for pseudo labelling and hard negative selection, builds upon a query generation approach called QGen [69], which applies a query generator trained on general out-of-domain data to generate in-domain queries for the document corpus. To allow finer analysis of the variables influencing the domain-adaptation performance of multilingual dense retrievers in a low-resource setup, we apply QGen without the GPL extension. However, we follow the GPL setup and thus introduce the method in section 4.2.1

Algorithm 2: Prompt-Based Query Generation

```
Input  : Document corpus $D = \{d_1, d_2, \ldots, d_n\}$, Set of
         prompts $\mathcal{P} = \{p_1, p_2\}$, Generative LLM $G$
$Q_{p_1}, Q_{p_2} = [], []$
foreach $d_i \in D$ do
    for $k \leftarrow 1$ to 3 do
        //Generate $k$-th queries using document $d_i$ and both
          prompts
        $q_k^{d_i,p_1} = G(d_i, p_1)$
        $q_k^{d_i,p_2} = G(d_i, p_2)$
        // Add generated queries to the respective query set
        $Q_{p_1} = Q_{p_1} \cup \{q_k^{d_i,p_1}\}$
        $Q_{p_2} = Q_{p_2} \cup \{q_k^{d_i,p_2}\}$
    end
end
Output : $Q_{p_1}, Q_{p_2}$
```

4.2 Pseudo Relevance Labelling

Knowledge distillation can be used to generate fine-grained relevance labels, enhancing the quality of the training data, and it can be combined with query generation methods. This combination possibility is beneficial since query generation methods tend to generate noisy data, and the pseudo-relevance scores can be used to filter it. Wang et al. [113] showed that combining query generation approaches with pseudo-relevance labels can substantially boost model performance. However, since generating pseudo-relevance labels is computationally expensive, we only use one method, comparing it to the performance of the generated datasets without relevance labels.

4.2.1 *Generative Pseudo Labelling*.

Wang et al. [113] suggested generative pseudo labelling (GPL), a state-of-the-art supervised query generation method, building on QGen [69] as mentioned in section 4.1.2. GPL [113] extends QGen by utilising what they call hard negatives instead of in-batch negatives and (pseudo-)labels generated by a cross-encoder instead of only coarse-grained relevance labels for model training. Generative Pseudo Labelling consists of three steps. First, a DocT5Query encoder-decoder generates three synthetic queries for every document. Then, two pre-trained dense retrievers each mine the 50 most similar documents for each generated query as hard negatives. In the third step, a pre-trained cross-encoder scores the positive and negative query-document pairs, generating fine-grained pseudo-relevance labels for the data. The cross-encoder scores are used as pseudo-labels:

$$\delta = CE(q, d_i^+) - CE(q, d_i^-) \tag{9}$$

Where:

- CE: cross-encoder score
- q: the query
- d_i^+: the positive document
- d_i^-: the negative document

The resulting relevance label δ is a logit ranging from $-Inf$ to $+Inf$. The higher the value, the more relevant the passage for the query. The resulting labelled training dataset is generated by uniformly sampling one negative and one positive document for each query.

We formalise the GPL task as:

$$\forall d_i \in D, Q_{labelled}^{d_i} = \{q_i, d_i^+, d_i^-, \delta_i\} \tag{10}$$

Where:

- D is the unlabelled document corpus
- $Q_{labelled}^{d_i^+}$ is the generated dataset containing the generated query q_i, the positive document d_i^+, the hard negative document d_i^- and the pseudo-relevance label δ_i for document d_i
- $q_i = G_{gen}(d_i)$ Where:
  - G_{gen} is the query generation model
- $d_i^+ = d_i$
- $d_i^- = G_{neg}(q_i, d_i^+)$ Where:
  - G_{neg} is the negative mining model
- $\delta_i = G_{cross}(q_i, d_i^+, d_i^-)$ Where:
  - G_{cross} is the pseudo-labelling cross-encoder model

The following figure summarises the GPL query generation procedure:

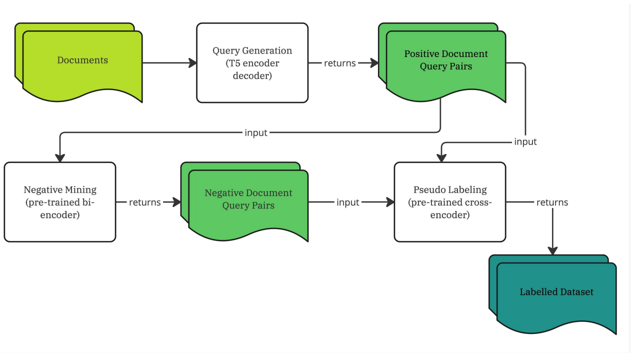
Documents
Query Generation (T5 encoder decoder)
returns
Positive Document Query Pairs
input
input
Negative Mining (pre-trained bi-encoder)
returns
Negative Document Query Pairs
input
Pseudo Labeling (pre-trained cross-encoder)
returns
Labelled Dataset

Figure 5: Generative Pseuo Labelling

We will follow this setup suggested by Wang et al. [113] to generate our second set of labelled training data. However, we will replace all models used in the original GPL code with multilingual models to adapt the approach to our multilingual setup. section 4.4.4 holds more details about our implementation.

4.3 Models

We fine-tune two multilingual dense retrievers for each generated German and English dataset. Since pre-trained Transformers require heavy computation to perform semantic search tasks, for example, finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations (65 hours) with BERT. We decided only to use sentence transformer models, as they reduce the time to fulfil the task to about 5 seconds [106]. We estimated 15 sentence transformer models with various architectures on our test data without further fine-tuning. We picked the two best-performing models of different sizes and architectures for fine-tuning. We measured performance using Precision@10, Recall@10, F1@10, MRR@10, Average Precision@10 and NDCG@10. In appendix A, We define these metrics. table 6 states the performance of the top five baseline models. The two best-performing models on all metrics for English, as well as German data, were a LLaMA-based model [108, 109] and a multilingual paraphrase-XML-R model [88, 90]. Both are single-representation bi-encoder[6]. Table 2 summarises their attributes:

The LLaMA-based model [108] is a lighter, more efficient model for generating sentence embeddings under constrained computational resources. In contrast, the XLM-R model [90] is more powerful and better equipped for multilingual data, with deeper layers, a larger hidden size and a more extensive vocabulary size. According to Ni et al. [80], larger models perform better on domain-adaptation tasks; we thus expect the XLM-R model to outperform the LLaMA-based model.

4.3.1 *Loss functions for model fine-tuning*.

We use the **Multiple Negatives Symmetric Ranking Loss** to fine-tune our models on the generated data without pseudo-relevance labels. It extends the **Multiple Negatives Ranking loss**

4.3.1.1 **Query-oriented Loss Function**

The query-oriented loss function is the exact negative log-likelihood (NLL) [48]. It maximises the probability of retrieving the relevant (positive) document d_i^+ for a given query q_i over any negative document $d' \in D^-$. (the set of irrelevant documents). We formalise this as:

$$L(q_i, d_i^+) = -log \frac{e^{f(\phi(q_i),\psi(d_i^+))}}{e^{f(\phi(q_i),\psi(d_i^+))} + \sum_{d' \in D^-} e^{f(\phi(q_i),\psi(d'))}} \tag{11}$$

Where:

- d_i^+ is a relevant document for query q_i
- $\phi(\cdot)$ and $\psi(\cdot)$ are the query and text encoder
- $f(\cdot)$ measures the similarity between query embedding $\phi(q_i)$ as well as text embedding $\psi(d_i^+)$
- D^- is the set of all documents except the positive one(s)

4.3.1.2 **Multiple Negatives Ranking Loss**

Since the Query-oriented Loss Function iterates over all indexed documents in the normalisation term computing, it is very time-consuming. The negative sampling trick was introduced to the negative log-likelihood to tackle that issue, leading to the Multiple Negatives Ranking Loss (MNRL) [48, 84]. It reduces computational costs compared to the exact formulation by sampling a set of negatives from all documents. Its objective can be summarised as increasing the likelihood of positive documents while decreasing the likelihood of sampled negative documents. We formalise the MNRL as:

$$L(q_i, d_i^+) = -log \frac{e^{f(\phi(q_i),\psi(d_i^+))}}{e^{f(\phi(q_i),\psi(d_i^+))} + \sum_{d' \in N_{q_i}} e^{f(\phi(q_i),\psi(d'))}} \tag{12}$$

Where:

[6] Since several studies [16, 21, 48, 51, 85, 91, 115, 119] on monolingual, English data, suggest improved domain adaptation if more complex retrieval systems combining retriever and re-ranker are used. We also tried integrating different cross-encoders for re-ranking, but since none improved the retrieval results, we used the two bi-encoders by themselves

Table 2: Model Overview

Aspect	LLaMA-based Embedding Model	Paraphrase-XLM-R-Multilingual-v1
Architecture	BERT-based 6 layers 12 attention heads	XLM-RoBERTa-based 12 layers 12 attention heads
Hidden Size	384	768
Model Size	22.7M parameters	278M parameters
Vocabulary Size	30,522	250,002
Max Sequence Length	256	128
Pooling Layer	Mean pooling (384-dim)	Mean pooling (768-dim)
Dropout	0.1 for attention and hidden layers	0.1 for attention and hidden layers
Special Features	Embedding normalization after pooling	N/A

- N_{q_i} is a small set of negative samples for the query q_i

It was derived from the InfoNCE loss [81], which contrasts a positive pair of examples with randomly sampled examples. If the loss uses in-batch negatives for a dense retriever trained in batches, the set of negatives N_{q_i} for the query q_i are all documents in the batch, apart from the positive document d_i^+

4.3.1.3 **Multiple Negatives Symmetric Ranking Loss**
To balance retrieval, the Multiple Negatives Symmetric Ranking Loss extends the MNRL by jointly optimising a query and a document-oriented loss function. The document-oriented loss function is the negative log-likelihood oriented at the document text [119]. It ensures that the relevant query q_i is ranked higher than negative queries $q^- \in Q^-$ and can be formalised as:

$$L_T(q_i, d_i^+) = -log \frac{e^{f(\psi(d_i^+),\phi(q_i))}}{e^{f(\psi(d_i^+),\phi(q_i))} + \sum_{q^- \in Q^-} e^{f(\psi(d_i^+),\phi(q^-))}} \tag{13}$$

Where:

- Q^- is a set of sampled negative queries.

Given this definition, the Multiple Negatives Symmetric Ranking Loss can be formalised as:

$$L_{MNSRL}(q_i, d_i^+) = L(q_i, d_i^+) + L_T(q_i, d_i^+) \tag{14}$$

This loss function ensures that queries and documents are optimised symmetrically for retrieval, using a smaller set of sampled negatives to maintain computational efficiency. Since we use generated queries, optimising query and document loss simultaneously is helpful as it integrates the data generation mechanism of the queries generated given the document into the loss formulation.

4.3.1.4 **MarginMSE Loss**
We utilise the MarginMSE loss [113] to optimise our models on the pseudo-labelled generative data. It improves the Multiple Negatives Ranking Loss, which only considers matching passages relevant. However, that cannot be guaranteed with synthetically generated queries and hard negative documents since the generated queries might not match the positive documents, and the retrieved negative documents might also be relevant to the query. We define the MarginMSE loss as:

$$L_{MargineMSE}(q_i, d_i^+, d_i^-, \delta_i) = -\frac{1}{M}\sum_{i=0}^{M-1} |\hat{\delta}_i - \delta_i|^2 \tag{15}$$

Where:

- δ_i is a relevance label
- M is the batch size
- $\hat{\delta}_i$ is the score margin of student dense retriever:
  - $\hat{\delta}_i = \phi(q_i)^T \psi(d_i^+) - \phi(q_i)^T \psi(d_i^-)$

Since we use cross-encoder scores as pseudo-relevance labels, we use the dot product as the similarity measurement due to the scores' infinite range.

4.4 Implementation

We ran all our experiments on a Tesla V100-PCIE-16GB GPU. Thus, optimising our code for memory use and efficiency was quite relevant. We used several checkpoints so that re-running the script does not re-do an already executed step.

4.4.1 *Data Pre-Processing*.

Before using the unlabelled and the test data, we remove duplicate documents as well as queries with the same ID and text, replace NAs with 0, remove leading and trailing whitespaces, newlines, and tabs from the text data, and, to ensure data consistency, we replace German umlauts so that ü becomes ue, ä becomes ae, ö becomes oe, and ß becomes ss. Since some documents contain repeated punctuation, which does not convey semantic meaning, we remove every punctuation mark occurring more than once (with a whitespace in between or not), except for the first one.

4.4.2 *Keyword Query Generation*.

We generally followed Mas et al.s [71] setup for the keyword generation. However, since our documents are shorter, containing fewer tokens and a smaller vocabulary than the documents they used, we adapted some thresholds based on the descriptive statistics of our unlabelled document corpora. For example, in the original code, a randomisation mechanism excluding short documents sets the token threshold to 100, defining documents containing less than 100 tokens as short. Our average document length is 10.9 tokens. Thus, We used a value of 10 instead. Ma et al. [71] also exclude stopwords during the tokenisation of the documents. Still, since they developed their code for a monolingual, English setting, we added stopwords specialised for the German language and used those for the keyword generation with the German unlabelled documents. Also, the original PROP [71] offers parallel processing on multiple workers. Since we adapted the method and only used one worker, we simplified the code by excluding the parts needed for parallel processing.

4.4.3 *Prompt-Based Query Generation*.

We constructed two prompts for the prompt-based query generation, each containing a description of the generative model's role and a more detailed task description. The two prompts contain varying amounts of information about the query to generate and the general nature of the document corpus. Prompt one is less informative, while prompt two contains query generation restrictions and information about the nature of the unlabelled document corpus.

The two English prompts P_1 and P_2 we use are:

P_1 =*Extract from the following text {text} {query_num} search queries. Consider the entire context, as it is crucial for understanding the text. The texts are from the context of chemical and pharmaceutical production environments. The queries need to be meaningful, as if you are supposed to use them to google. A query should contain between 1 to 7 words. Reply with a list of queries separated by semicolon. Keep only the text of queries, no enumeration.*

P_2 =*Generate {query_num} search queries for the following text {text}. The queries need to be meaningful as if you are supposed to use them to google. A query should contain between 1 to 7 words. Minimise using tokens with digits. Avoid using persons names. Reply with a list of strings where each string is a query, the queries need to be separated by a semicolon. Keep only the text of queries, no enumeration. Consider the entire context, as it is crucial for understanding the text. The texts are logs from a chemical production factory.*

The two German prompts are:

P_1 =*Entnimm aus dem folgenden Text {text} {query_num} Suchanfragen. Berücksichtige dabei den gesamten Kontext, da er für das Verständnis des Textes entscheidend ist. Die Texte stammen aus dem Kontext der chemischen und pharmazeutischen Produktionsumgebungen. Die Anfragen müssen sinnvoll sein, als ob sie zur Google-Suche verwendet werden sollen. Eine Anfrage sollte zwischen 1 bis 5 Wörter enthalten. Antwort mit einer Liste von Anfragen, getrennt durch Semikolon. Behalte nur den Text der Anfragen, keine Aufzählung.*

P_2 =*Generiere {query_num} Suchanfragen für den folgenden Text 'text'. Die Anfragen müssen sinnvoll sein, als würdest du sie zum Googlen verwenden. Eine Anfrage sollte zwischen 1 und 5 Wörtern enthalten. Minimiere die Verwendung von Token mit Ziffern. Verwende keine Namen von Personen. Antworte mit einer Liste von Zeichenketten, wobei jede Zeichenkette eine Anfrage darstellt, die Anfragen sollen durch ein Semikolon voneinander getrennt sein. Behalte nur den Text der Anfragen, keine Aufzählung. Berücksichtige den gesamten Kontext, da er für das Verständnis des Textes entscheidend ist. Die Texte sind Logs aus einer Firma im Bereich chemische Prozessindustrie.*

During query generation, we set {query_num} to 3 and replaced {text} with the text of the unlabelled document. We analyse the effects of using two different prompt-s in section 5.1

4.4.4 *GPL*.

Since the GPL [113] train method includes training a model on the generated dataset, We reimplemented only its query generation, negative selection and pseudo-labelling parts. We also replaced the monolingual English models with multilingual models capable of processing German and English data. We tried to find a model similar to the initially used monolingual English model architecture. We replaced the T5-based model for query generation with a multilingual T5 model fine-tuned on the MMARCO dataset for query generation [2]. To replace the ms-marco-MiniLM-L-6-v2 cross-encoder [1] used for pseudo labelling the generated data, we used the msmarco-MiniLM-L6-en-de-v1 cross-encoder [1]. For the negative selection, we use the two best-performing baseline models [90, 108].

4.4.5 *Indexing*.

Since the amount of data we need to search during inference is relatively small (20,500 documents), we do not index it. Instead, we keep the embeddings in memory and perform an exhaustive search, computing the cosine similarity of the query and all text vectors to select the top documents for a query.

4.4.6 *Model Fine-tuning*.

We experimentally set our learning rate for each query generation method and model combination. For the in-batch negative training with Multiple Negatives Symmetric Ranking Loss[7], we use the most significant possible batch sizes to increase the negative document distribution, as the batch size directly determines the number of available negatives. For the smaller LLaMA-based model [108], that was a batch size of 200, and for the larger XLM-R Model [90], a batch size of 80.

For the training on the pseudo-labelled datasets with MarginMSE loss, we used a fixed batch size of 32 as suggested by Wang et al. [113].

For both negative selection approaches, we set the warmup ratio to 0.1 and used the default AdamW optimiser, which includes learning rate scheduling. Also, we fine-tuned our models using an Early Stopping mechanism, measuring precision@3 on a validation set with a patience of five. For training with Multiple Negatives, Symmetric Ranking Loss, and in-batch negatives, the validation set consists of 20% of the generated training data for the specific query method and relevance label combination. For training with MarginMSE loss, we set the size of the evaluation data depending on the data generation method used since that influences the dataset size. For the GPL-generated pseudo-relevance labelled data with 1,120,000 examples, we keep it at 20%; for all other query generation methods with 4,480,000 pseudo-labelled training examples, we reduce the validation dataset size to 1% of the training examples since a larger amount of validation data exceeds the memory capacities of our GPU. For all validation datasets independent of query generation mechanism and relevance label, we added 10,000 unlabelled documents to the validation dataset. We chose precision@3 as an evaluation metric to leave a margin for errors such as false negatives, even though we generated three distinct queries for every document with every query generation method; thus, in theory, each query should only match one relevant document in the whole corpus. Since we used this early stopping mechanism, we set the number of Epochs to 500, evaluated the model, and saved it every 500 steps.

[7] We also tried MegaBatchMarginLoss, GISTEmbedLoss and CachedGISTEmbedLoss [103], for fine-tuning the models with in-batch negatives, however, a batch size >= 500 as well as holding two models in memory exceeded the capacities of our GPU

5 EXPERIMENTS

5.1 General Experimental Set-Up

Figure 6 visualises our experimental set up.

We highlighted the experimental variables in different colours. However, for simplicity, we excluded language as a variable. We separately apply the complete experimental setup for the English and German unlabelled documents. The three query generation methods on the left are highlighted in yellow, the pseudo labelling in the middle in green, and the two different sentence transformers on the right in blue.

The evaluation of our experiments involves fine-tuning the two models on the differently generated datasets. After fine-tuning, we measure their performance on the test set reporting Precision@10, Recall@10, F1@10, MRR@10, MAP@10, and NDCG@10. Appendix A contains a definition of the metrics.

In total, we generated twelve different datasets. Three were generated by the different query generation methods without relevance labels, three for each query method with pseudo-relevance labels and hard negatives and that times two since the datasets exist separately for German and English data. Thus, we trained 12 models in total since we trained the models on all datasets for all languages.

5.2 Results

Table 3 provides a comprehensive evaluation of the different models trained on English data. Table 4 summarises the results for the German models. Each is structured according to our explorative variables. The three distinct data generation strategies: **Keyword**, **Prompt** or **QGen**. The use of either **no pseudo labels** or **fine-grained pseudo-relevance labels (GPL)** and the two different models: the **LLaMA-based** embedding model and the **XLM-R** based. We analyse the influence of each experimental variable in the following subsections.

5.3 Languages

We start with the variable language since it has the most prevalent influence on model performance.

5.3.1 *English Data*.

The results for the English Data are summarised in table 3. As indicated by the bold number, the best-performing model for English data, on average across all metrics, was the XLM-R baseline model. Thus, our domain adaptation approach did not lead to the retrieval of more relevant documents or a better document ranking. We discuss possible reasons in section 5.3.3, but for the further analysis of the other three variables we explored - query generation method, pseudo-relevance labelling and model size - we will focus solely on the German retrieval performance.

5.3.2 *German Data*.

A German model reached the overall best average performance of 71.65% across all metrics. Our domain adaptation strategy thus **improved the retrieval performance by 3.58%**. The combination of domain-adaptation variables to create that model is: Fine-tuning on QGen data [8] without pseudo-relevance labels, using the XLM-R architecture model with a larger model size compared to the LLaMA-based model. In the following sections, we analyse the influence of the other variables on target domain retrieval performance in detail.

5.3.3 *Discussion*.

Even though the best-performing model is German, all other German models are outperformed by the English baseline XLM-R model, even though the domain adaptation approaches lead to substantial performance improvements on the German testset. That allows us to formulate some theories about why the domain adaptation approaches only improved retrieval for the German but not the English data.

The LLaMA-based [108] and the XLM-R [90] model demonstrate strong performance in the English baseline configuration. For instance, XLM-R's baseline shows high scores across the board, especially with metrics like Precision@10 (89.50%) and F1@10 (54.70%). This strong performance may indicate that these models already capture relevant information, and further domain-specific adaptation fails to yield substantial improvements. Multilingual models are primarily trained on a significantly larger and more diverse set of English than German data. This might also be true for the LLaMA-based and the XLM-R model [114]. English is more commonly represented in multilingual pre-training corpora, which means that multilingual models are better optimised for English retrieval tasks, in general [121]. This helps explain why the baseline models perform better on English test data. Thus, the effectiveness of domain adaptation might depend on language. Domain adaptation methods, such as keyword-based or prompt-based training, may be more effective in German because the baseline model had more room for improvement. While English models already perform well due to better pre-training and more abundant data, the German baseline performance, especially of the XLM-R model, was lower, leaving more opportunity for adaptation methods to make a noticeable difference. The improvement in retrieval performance for the German data after domain adaptation reflects how specialised training helps models overcome some inherent challenges of handling non-English languages in multilingual settings. However, the overall worse performance of German models compared to English ones is primarily due to the stronger baseline for English and the larger amount of training data available for English. Future work could involve fine-tuning with larger, domain-specific German corpora or monolingual German models to close the performance gap between languages.

[8] thus, GPL generated data but only the query document pairs without the relevance labels or hard negatives

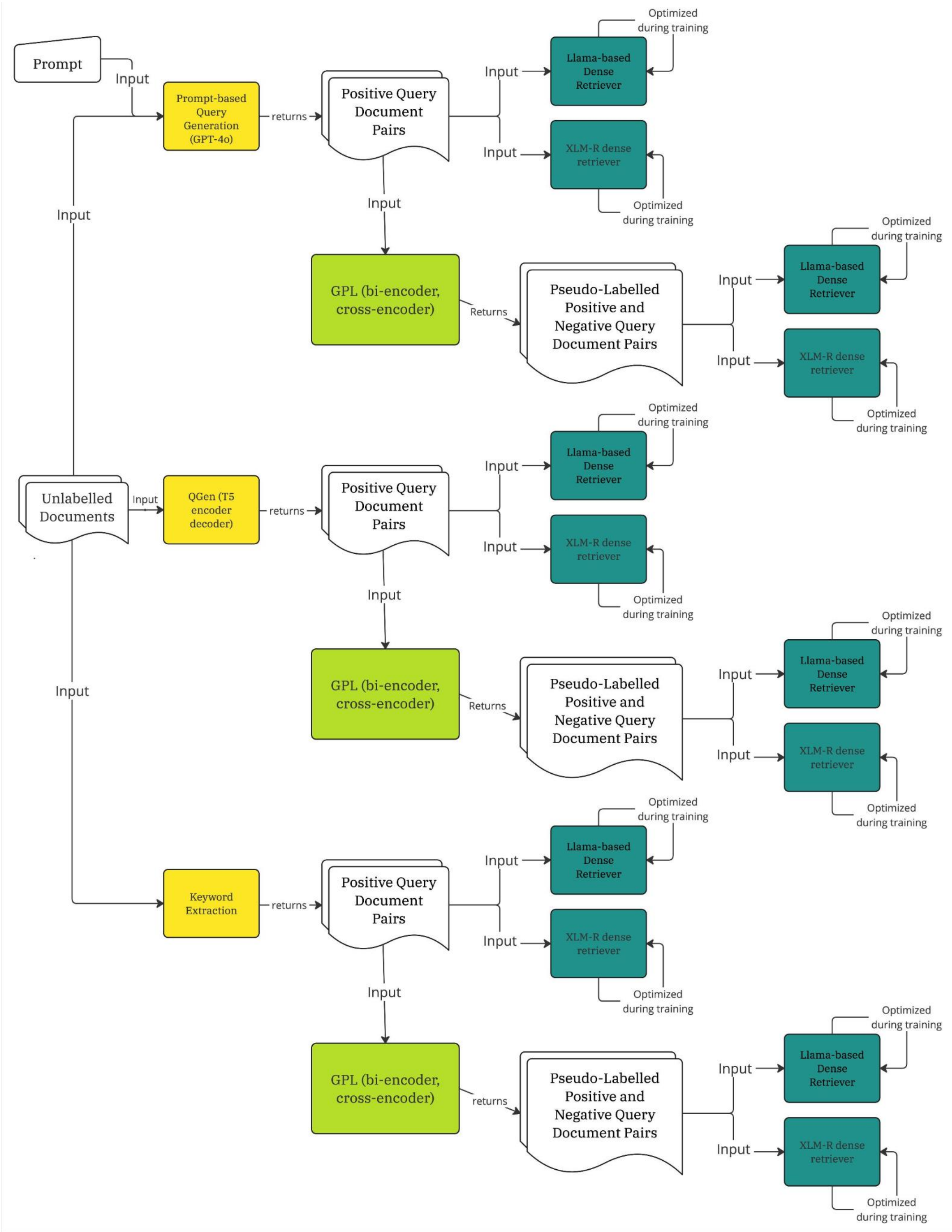
Prompt
Input
Prompt-based Query Generation (GPT-4o)
returns
Positive Query Document Pairs
Input
Llama-based Dense Retriever
Optimized during training
Input
XLM-R dense retriever
Optimized during training
Input
Input
GPL (bi-encoder, cross-encoder)
Returns
Pseudo-Labelled Positive and Negative Query Document Pairs
Input
Llama-based Dense Retriever
Optimized during training
Input
XLM-R dense retriever
Optimized during training
Unlabelled Documents
Input
QGen (T5 encoder decoder)
returns
Positive Query Document Pairs
Input
Llama-based Dense Retriever
Optimized during training
Input
XLM-R dense retriever
Optimized during training
Input
GPL (bi-encoder, cross-encoder)
Returns
Pseudo-Labelled Positive and Negative Query Document Pairs
Input
Llama-based Dense Retriever
Optimized during training
Input
XLM-R dense retriever
Optimized during training
Input
Keyword Extraction
returns
Positive Query Document Pairs
Input
Llama-based Dense Retriever
Optimized during training
Input
XLM-R dense retriever
Optimized during training
Input
GPL (bi-encoder, cross-encoder)
returns
Pseudo-Labelled Positive and Negative Query Document Pairs
Input
Llama-based Dense Retriever
Optimized during training
Input
XLM-R dense retriever
Optimized during training

Figure 6: Experimental Set Up

5.4 Query Generation Methods and Pseudo Relevance Labelling

5.4.1 Generated data.

The following table summarises the amount of queries generated by each query generation method for both German and English:

The Keyword query method generated 32,584 German and 54,930 English queries. This approach produced the fewest queries overall, particularly for the German dataset. The difference between languages arises since documents in the unlabelled German corpus contain an average of 12.236 tokens, with a total vocabulary of 8.429. In contrast, records in the unlabelled English corpus contain

Table 3: English Data, Performance on Testset

English			**Precision@10**	**Recall@10**	**F1@10**	**MRR@10**	**aP_N@10**	**ndcg@10**	**AVG**
LLaMA-based	baseline		84.70%	41.70%	52.30%	97.40%	87.00%	51.50%	69.10%
	keyword	no labels	81.55%	42.45%	52.11%	98.25%	93.47%	52.71%	70.09%
		pseudo-relevance labels	84.14%	44.26%	54.22%	96.49%	92.33%	53.02%	70.74%
	QGen	no labels	83.79%	42.75%	52.73%	96.20%	91.35%	51.27%	69.68%
		pseudo-relevance labels (GPL)	83.23%	42.88%	52.92%	93.71%	91.92%	51.34%	69.33%
	prompt-1	no labels	83.07%	44.53%	54.09%	96.64%	93.65%	56.00%	71.33%
		pseudo-relevance labels	84.41%	44.23%	54.29%	93.86%	91.88%	53.34%	70.33%
	prompt-2	no labels	82.31%	42.82%	52.65%	97.37%	93.35%	52.94%	70.24%
		pseudo-relevance labels	80.63%	40.51%	50.27%	96.05%	92.32%	49.84%	68.27%
XLM-R	baseline		89.50%	43.40%	54.70%	100.00%	90.30%	51.30%	**71.53**%
	keyword	no labels	78.47%	41.24%	50.60%	97.08%	92.85%	50.96%	68.53%
		pseudo-relevance labels	66.57%	36.91%	43.89%	93.04%	82.26%	44.91%	61.26%
	QGen	no labels	77.97%	39.80%	49.03%	100.00%	92.23%	49.38%	68.07%
		pseudo-relevance labels (GPL)	76.31%	38.33%	47.84%	96.78%	90.03%	47.30%	66.10%
	prompt-1	no labels	80.66%	41.97%	51.53%	96.20%	92.27%	51.61%	69.04%
		pseudo-relevance labels	63.96%	33.38%	41.15%	91.80%	83.34%	42.28%	59.32%
	prompt-2	no labels	80.16%	41.61%	51.17%	98.25%	93.07%	51.39%	69.27%
		pseudo-relevance labels	74.63%	37.95%	46.60%	97.37%	89.15%	47.76%	65.58%

Table 4: German Data, Performance on Testset

German			**Precision@10**	**Recall@10**	**F1@10**	**MRR@10**	**aP_N@10**	**ndcg@10**	**AVG**
LLaMA-based	baseline		71.10%	54.30 %	**54.30**%	90.40%	82.10%	56.20%	68.07%
	keyword	no labels	67.73%	47.40%	46.02%	83.33%	80.67%	51.77%	62.82%
		pseudo-relevance labels	70.38%	50.27%	47.74%	85.59%	83.39%	55.00%	65.40%
	QGen	no labels	71.05%	53.07%	49.16%	90.64%	89.93%	59.79%	68.94%
		pseudo-relevance labels (GPL)	71.58%	53.52%	49.36%	94.74%	91.28%	59.29%	69.96%
	prompt-1	no labels	71.46%	50.85%	49.31%	88.01%	85.77%	58.26%	67.28%
		pseudo-relevance labels	70.20%	49.89%	47.14%	90.35%	88.98%	55.61%	67.03%
	prompt-2	no labels	71.22%	50.77%	48.96%	89.47%	87.20%	59.18%	67.80%
		pseudo-relevance labels	70.73%	51.73%	48.07%	94.74%	91.36%	57.79%	69.07%
XLM-R	baseline		66.30%	52.80%	44.00%	92.10%	79.80%	53.90%	64.82%
	keyword	no labels	70.28%	49.80%	47.65%	85.57%	84.39%	51.88%	64.93%
		pseudo-relevance labels	58.29%	43.26%	39.46%	84.80%	81.44%	46.36%	58.94%
	QGen	no labels	**72.82**%	**57.97** %	50.90 %	95.09%	91.62%	**61.53**%	**71.65**%
		pseudo-relevance labels (GPL)	69.85%	54.12%	48.67%	**95.61** %	92.07%	58.94%	69.88%
	prompt-1	no labels	71.37%	54.71%	49.58%	92.31%	89.71%	59.72%	69.57%
		pseudo-relevance labels	69.91%	50.83%	47.82%	89.47%	86.77%	55.07%	66.65%
	prompt-2	no labels	71.96%	55.13%	49.87%	93.42%	90.85%	61.15%	70.39%
		pseudo-relevance labels	71.70%	57.26%	49.82%	93.89%	**92.31**%	61.01%	71.00%

Table 5: Amount of Generated Queries

Query Generation Methods	German Queries	English Queries
Keyword queries	32 584	54 930
GPL	59 912	53 864
Prompt 1 queries	95 324	94 009
Prompt 2 queries	81 260	72 674

an average of 95.134 tokens and a total vocabulary of 13.126 tokens. This impacts the results of the keyword generation method since it relies on fixed thresholds for the number of tokens in a document to qualify it for query generation. The larger average document size and vocabulary in the English document corpus, thus, lead to more English keyword-generated queries than in German.

GPL generated 59,912 German queries and 53,864 English Queries, showing a more balanced output across both languages than the other query generation methods.

The first prompt generates the most prompt-based queries overall, with 95,324 for German and 94,009 for English. The queries generated by the second prompt are also substantial: 81,260 German queries and 72,674 English queries. However, this output is somewhat lower than the number generated by the first prompt. This method's high output can be attributed to challenges in parallel processing of the two prompts and the caching mechanism. This leads to duplicated queries across seeds. Also, the generative model sometimes returns more than the specified number of queries for a document.

Regarding language differences, all methods produce more German than English queries, except the extractive approach.

The GPL dataset consists of 1,120,000 labelled examples for the pseudo-labelled generated data, while all other datasets consist of 4,480,000 labelled examples.

5.4.2 *Model Performance - German data*.

The keyword-based approach without pseudo-relevance labels performs worse than the baseline models on most metrics. For example, for the LLaMA-based model, Precision@10 drops to 67.73% (from 71.10%), and the average score falls to 62.82%. Even when pseudo-relevance labels are used, the average score only improves to 65.40%, which is still below the baseline. However, for the XLM-R models, adding pseudo-relevance scores to the keyword generation leads to sharp performance drops. This indicates that the keyword-based method, especially with pseudo-relevance labels, does not align well with the retrieval task. The bad performance might also be attributed to the small amount of German queries generated by the keyword approach, thus leading to fewer training examples.

The prompt-based approach performs better than the keyword-based method but seldom surpasses the QGen approach. Thus, prompt-based methods offer some improvement, likely because the prompts help guide the model in generating more domain-specific queries. However, that improvement is marginal for the LLaMA-based model. At the same time, Prompt-based methods work well with XLM-R, indicating an interaction between model size or architecture and this query generation approach. For XLM-R, the second, more complex prompt also yields significantly better retrieval performance results, indicating that adding knowledge about the needed query structure and the unlabeled document corpus enables the generative model to create better prompts. This finding supports the findings introduced by Dai et al. [23], which also found that enriching the prompts with further in-domain knowledge improves retrieval performance. However, they used labelled data examples to do so.

Both LLaMA and XLM-R show that QGen consistently outperforms keyword-based and prompt-based approaches, with XLM-R benefiting the most. This suggests that the t5-encoder-decoder model generates the semantically richest queries. However, a quick analysis of the QGEN Queries revealed they are most similar in structure to actual questions compared to the keyword and prompt-generated queries. Thus, the QGen queries might be more similar to the source-domain data used to train the two dense retrievers, facilitating knowledge transfer to the new domain [92, 132].

5.5 Models - German Data

In comparing the two models, LLaMA-based and XLM-R, across the German retrieval task, the results indicate some interesting differences in their performance, particularly in relation to the query generation methods and overall performance. The LLaMA-based model's baseline performance is stronger than that of XLM-R. It achieves a higher average score (68.07%) compared to XLM-R's 64.82%. This suggests that the LLaMA-based model is better out-of-the-box for German retrieval tasks, likely due to better representation of domain-specific information and a stronger initial language understanding. However, the performance gains from query generation methods are generally more incremental for the LLaMA model, while the XLM-R model shows more significant improvements when enhanced through fine-tuning on generated target domain data. The effect is fascinating since the different architecture and larger size of the XLM-R model are probably related to the more effective fine-tuning, supporting the findings of Ni et al. [80].

6 CONCLUSION

In this thesis, we addressed the research question: "How to perform domain adaptation for multilingual semantic search in a low-resource setup?" We experimentally explored domain adaptation for multilingual semantic search in a zero-shot setting, using unlabelled German and English documents from the chemical process industry alongside a small labelled test set for evaluation. We examined the impact of four key variables on domain adaptation in a multilingual context.

Firstly, we assessed different query generation methods, including an extractive approach that samples keyword queries, a generative approach using a multilingual T5-based model, and a prompt-based method using GPT-4o.

Secondly, we evaluated the role of pseudo-labelling through knowledge distillation to address potential issues with the quality of generated queries, following the approach suggested by Wang et al. [113].

Thirdly, we studied the influence of model size by fine-tuning a smaller LLaMA-based embedding model and a larger XLM-RoBERTa model, investigating their respective capacities for domain adaptation.

Finally, we considered the effect of dataset language, analysing performance for English and German test data.

By systematically analysing these variables, we aimed to identify effective strategies for domain adaptation in multilingual low-resource settings.

The explored domain adaptation approaches for the English setup did not yield performance gains, probably because multilingual models were already strong baseline performers on English data, leaving less room for improvement.

For the German data, we conclude that effective domain adaptation for multilingual semantic search in a low-resource setup requires careful query generation and model architecture optimisation.

The XLM-R model, though initially weaker, showed significant improvements with advanced query generation techniques, like QGen, on German test data. In contrast, the LLaMA-based model had stronger baseline performance but responded less to query generation enhancements. This indicates a possible interaction between model size or architecture and the effectiveness of domain adaptation approaches that rely on fine-tuning the model with generated in-domain data. With the larger model showing more adaptability to domain-specific content.

Additionally, the use of pseudo-labelling had limited impact. Thus, successful domain adaptation in low-resource multilingual settings requires a combination of well-designed query generation strategies and exemplary model architectures, focusing on model scalability and query informativeness.

7 OUTLOOK

This thesis is only a tiny step towards adapting multilingual dense retrievers to low-resource domains. For future research, exploring the unified optimisation of a multilingual dense retriever on generated labelled data for multiple languages simultaneously is an interesting direction. Another is further exploring the use of unlabelled data generated in day-to-day industry processes for generating labelled datasets through query generation methods.

REFERENCES

[1] 2021. *msmarco-MiniLM-L6-en-de-v1 cross encoder.* https://huggingface.co/cross-encoder/msmarco-MiniLM-L6-en-de-v1

[2] 2021. *mt5-small-german-query-generation query generator.* https://huggingface.co/ml6team/mt5-small-german-query-generation

[3] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive Multi-task Representations with Pre-Finetuning. Online. arXiv:2101.11038

[4] Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing and Management* 39, 1 (jan 2003), 45–65. https://doi.org/10.1016/s0306-4573(02)00021-3

[5] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. Online. arXiv:1906.05416

[6] Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics* 7 (nov 2019), 597–610. https://doi.org/10.1162/tacl_a_00288

[7] Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual Open-Retrieval Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 547–564. https://doi.org/10.18653/v1/2021.naacl-main.46

[8] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. Online. arXiv:1611.09268

[9] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Data Augmentation for Information Retrieval using Large Language Models. Online. arXiv:2202.05144

[10] Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2021. mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset. Online. arXiv:2108.13897

[11] Anna Bringmann and Anastasia Zhukova. 2024. Domain Adaptation of Multilingual Semantic Search – Literature Review. arXiv:2402.02932 [cs.IR]

[12] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. Online. arXiv:2005.14165

[13] Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic Re-tuning with Contrastive Tension. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Ov_sMNau-PF

[14] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation. Online. arXiv:1708.00055

[15] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. arXiv:2002.03932

[16] Tao Chen, Mingyang Zhang, Jing Lu, Michael Bendersky, and Marc Najork. 2022. Out-of-Domain Semantics to the Rescue! Zero-Shot Hybrid Retrieval Models. In *Lecture Notes in Computer Science*. Springer International Publishing, 95–110. https://doi.org/10.1007/978-3-030-99736-6_7

[17] Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen tau Yih. 2022. Salient Phrase Aware Dense Retrieval: Can a Dense Retriever Imitate a Sparse One? Online. arXiv:2110.06918

[18] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3397271.3401043

[19] Muthu Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Association for Computational Linguistics, Florence, Italy, 250–259. https://doi.org/10.18653/v1/W19-4330

[20] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Online. arXiv:1409.1259

[21] Zhuyun Dai and Jamie Callan. 2020. Context-Aware Term Weighting For First Stage Passage Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1533–1536. https://doi.org/10.1145/3397271.3401204

[22] Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog Inpainting: Turning Documents into Dialogs. Online. arXiv:2205.09073

[23] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot Dense Retrieval From 8 Examples. Online. arXiv:2209.11755

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Online. https://doi.org/10.48550/ARXIV.1810.04805 arXiv:1810.04805

[25] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, and Jiafeng Guo. 2022. Pre-training Methods in Information Retrieval. arXiv:2111.13853 [cs.IR]

[26] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 878–891. https://doi.org/10.18653/v1/2022.acl-long.62

[27] Hengxin Fun, Sunil Gandhi, and Sujith Ravi. 2021. Efficient Retrieval Optimized Multi-task Learning. Online. arXiv:2104.10129

[28] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. Online. arXiv:1409.7495

[29] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. Online. arXiv:1505.07818

[30] Luyu Gao and Jamie Callan. 2021. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. Online. arXiv:2108.05540

[31] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. Online. arXiv:2104.08821

[32] A Gretton, AJ Smola, J Huang, M Schmittfull, and B Borgwardt, KM.and Schölkopf. 2009. *Covariate shift and local learning by distribution matching*. MIT Press, Cambridge, MA, USA, 131–160.

[33] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2017. A Deep Relevance Matching Model for Ad-hoc Retrieval. Online. arXiv:1711.08611

[34] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2019. A Deep Look into Neural Ranking Models for Information Retrieval. Online. arXiv:1903.06902

[35] Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective Parallel Corpus Mining using Bilingual Sentence Embeddings. In *Proceedings of the Third Conference on Machine Translation:*

Research Papers. Association for Computational Linguistics, Brussels, Belgium, 165–176. https://doi.org/10.18653/v1/W18-6317

[36] Michael A. Hedderich, David Adelani, Dawei Zhu, Jesujoba Alabi, Udia Markus, and Dietrich Klakow. 2020. Transfer Learning and Distant Supervision for Multilingual Transformer Models: A Study on African Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 2580–2591. https://doi.org/10.18653/v1/2020.emnlp-main.204

[37] Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. Online. arXiv:2010.12309

[38] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. Online. arXiv:1705.00652

[39] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (nov 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[40] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2021. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. Online. arXiv:2010.02666

[41] Sebastian Hofstätter, Omar Khattab, Sophia Althammer, Mete Sertkan, and Allan Hanbury. 2022. Introducing Neural Bag of Whole-Words with ColBERTer: Contextualized Late Interactions using Enhanced Reduction. Online. arXiv:2203.13088

[42] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. Online. arXiv:2104.06967

[43] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. Online. arXiv:1902.00751

[44] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*. ACM Press. https://doi.org/10.1145/2505515.2505665

[45] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. Online. arXiv:1905.01969

[46] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Herve Jegou, and Edouard Grave. 2018. Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion. Online. arXiv:1804.07745

[47] T. Joyce and R. M. Needham. 1958. The thesaurus approach to information retrieval. *American Documentation* 9, 3 (aug 1958), 192–197. https://doi.org/10.1002/asi.5090090305

[48] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[49] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. Online. arXiv:2004.12832

[50] Weize Kong, Swaraj Khadanga, Cheng Li, Shaleen Kumar Gupta, Mingyang Zhang, Wensong Xu, and Michael Bendersky. 2022. Multi-Aspect Dense Retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 3178–3186. https://doi.org/10.1145/3534678.3539137

[51] Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach. Online. arXiv:2010.01195

[52] Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4483–4499. https://doi.org/10.18653/v1/2020.emnlp-main.363

[53] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning Dense Representations of Phrases at Scale. Online. arXiv:2012.12624

[54] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. Online. arXiv:1906.00300

[55] Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them. Online. arXiv:2102.07033

[56] Hang Li. 2011. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00348ED1V01Y201104HLT012

[57] Minghan Li and Eric Gaussier. 2022. Domain Adaptation for Dense Retrieval through Self-Supervision by Pseudo-Relevance Labeling. Online. arXiv:2212.06552

[58] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. Online. arXiv:2101.00190

[59] Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based Zero-shot Retrieval through Query Generation. Online. arXiv:2009.10270

[60] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling Dense Representations for Ranking using Tightly-Coupled Teachers. Online. arXiv:2010.11386

[61] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2021. On Cross-Lingual Retrieval with Multilingual Text Encoders. Online. arXiv:2112.11031

[62] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. Online. arXiv:2101.00190

[63] Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-14267-3

[64] Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022. Collaborative Filtering with Attribution Alignment for Review-based Non-overlapped Cross Domain Recommendation. Online. arXiv:2202.04920

[65] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too. Online. arXiv:2103.10385

[66] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Online. arXiv:1907.11692

[67] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. Online. arXiv:1502.02791

[68] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345. https://doi.org/10.1162/tacl_a_00369

[69] Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation. Online. arXiv:2004.14503

[70] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3477495.3531772

[71] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM. https://doi.org/10.1145/3437963.3441777

[72] Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2019. Universal Text Representation from BERT: An Empirical Study. Online. https://doi.org/10.48550/ARXIV.1910.07973 arXiv:1910.07973

[73] Sean MacAvaney, Kai Hui, and Andrew Yates. 2017. An Approach for Weakly-Supervised Deep Information Retrieval. In *SIGIR 2017 Workshop on Neural Information Retrieval*. https://arxiv.org/abs/1707.00189v2

[74] Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019. Content-Based Weak Supervision for Ad-Hoc Re-Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3331184.3331316

[75] Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen tau Yih, Barlas Oğuz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task Retrieval for Knowledge-Intensive Tasks. Online. arXiv:2101.00117

[76] Zhuoyuan Mao, Chenhui Chu, and Sadao Kurohashi. 2022. EMS: Efficient and Effective Massively Multilingual Sentence Representation Learning. Online. arXiv:2205.15744

[77] Zhuoyuan Mao and Tetsuji Nakagawa. 2023. LEALLA: Learning Lightweight Language-agnostic Sentence Embeddings with Knowledge Distillation. Online. arXiv:2302.08387

[78] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting Similarities among Languages for Machine Translation. Online. arXiv:1309.4168

[79] Bhaskar Mitra and Nick Craswell. 2017. Neural Models for Information Retrieval. Online. arXiv:1705.01509

[80] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hern'andez 'Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large Dual Encoders Are Generalizable Retrievers. In *Conference on Empirical Methods in Natural Language Processing*.

[81] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. Online. arXiv:2004.09930

[82] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[83] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. Online. arXiv:1904.07531

[84] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 5835–5847. https://doi.org/10.18653/v1/2021.naacl-main.466

[85] Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. Learning to Retrieve Passages without Supervision. Online. arXiv:2112.07708

[86] Alan Ramponi and Barbara Plank. 2020. Neural Unsupervised Domain Adaptation in NLP—A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 6838–6855. https://doi.org/10.18653/v1/2020.coling-main.603

[87] Revanth Gangi Reddy, Vikas Yadav, Md Arafat Sultan, Martin Franz, Vittorio Castelli, Heng Ji, and Avirup Sil. 2021. Towards Robust Neural Retrieval Models with Synthetic Pre-Training. Online. arXiv:2104.07800

[88] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Online. arXiv:1908.10084

[89] Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4512–4525. https://doi.org/10.18653/v1/2020.emnlp-main.365

[90] Nils Reimers and Iryna Gurevych. 2024. *paraphrase-xlm-r-multilingual-v1 embedding model*. https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1

[91] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2023. A Thorough Examination on Zero-shot Dense Retrieval. Online. arXiv:2204.12755

[92] Mehdi Rezagholizadeh, Aref Jafari, Puneeth S.M. Saladi, Pranav Sharma, Ali Saheb Pasand, and Ali Ghodsi. 2022. Pro-KD: Progressive Distillation by Following the Footsteps of the Teacher. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 4714–4727. https://aclanthology.org/2022.coling-1.418

[93] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation* 60, 5 (oct 2004), 503–520. https://doi.org/10.1108/00220410410560582

[94] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[95] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994 (NIST Special Publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), 109–126. http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[96] Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2023. Improving Passage Retrieval with Zero-Shot Question Generation. Online. arXiv:2204.07496

[97] Gerard Salton. 1962. Some experiments in the generation of word and document associations. In *Proceedings of the December 4-6, 1962, fall joint computer conference on - AFIPS '62 (Fall)*. ACM Press. https://doi.org/10.1145/1461518.1461544

[98] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (jan 1988), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

[99] G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (nov 1975), 613–620. https://doi.org/10.1145/

361219.361220

[100] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Online. arXiv:1910.01108

[101] Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 5445–5460. https://doi.org/10.18653/v1/2020.emnlp-main.439

[102] Xiaoyu Shen, Svitlana Vakulenko, Marco del Tredici, Gianni Barlacchi, Bill Byrne, and Adrià de Gispert. 2022. Low-Resource Dense Retrieval for Open-Domain Question Answering: A Comprehensive Survey. Online. arXiv:2208.03197

[103] Aivin V. Solatorio. 2024. GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning. arXiv:2402.16829 [cs.LG] https://arxiv.org/abs/2402.16829

[104] Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. 2021. Few-Shot Text Ranking with Meta Adapted Synthetic Weak Supervision. Online. arXiv:2012.14862

[105] Weng Lam Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Xingjian Zhang, Yuxiao Dong, Jiahua Liu, Maodi Hu, and Jie Tang. 2022. Parameter-Efficient Prompt Tuning Makes Generalized and Calibrated Neural Text Retrievers. Online. arXiv:2207.07087

[106] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. Online. arXiv:2010.08240

[107] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Proceedings of the 2021 Neural Information Processing Systems (NeurIPS-2021): Track on Datasets and Benchmarks*. https://arxiv.org/abs/2104.08663

[108] thuan ton. 2024. *Llama embedding model*. https://huggingface.co/thuan9889/llama_embedding_model_v1

[109] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[110] Brandon Tran, Maryam Karimzadehgan, Rama Kumar Pasumarthi, Mike Bendersky, and Don Metzler. 2019. Domain Adaptation for Enterprise Email Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[111] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. Online. arXiv:1412.3474

[112] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. Online. arXiv:1706.03762

[113] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2345–2360. https://doi.org/10.18653/v1/2022.naacl-main.168

[114] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. *arXiv preprint arXiv:2402.05672* (2024).

[115] Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. BERT-Based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval* (Virtual Event, Canada) *(ICTIR '21)*. Association for Computing Machinery, New York, NY, USA, 317–324. https://doi.org/10.1145/3471158.3472233

[116] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL] https://arxiv.org/abs/2109.01652

[117] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3077136.3080809

[118] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. Online. arXiv:2007.00808

[119] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. LaPraDoR: Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval. Online. arXiv:2203.06169

[120] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual Universal Sentence Encoder for Semantic Retrieval. Online. arXiv:1907.04307

[121] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual Universal Sentence Encoder for Semantic Retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, 87–94. https://doi.org/10.18653/v1/2020.acl-demos.12

[122] Changlong Yu, Hongming Zhang, Yangqiu Song, and Wilfred Ng. 2022. CoCoLM: COmplex COmmonsense Enhanced Language Model with Discourse Relations. Online. arXiv:2012.15643

[123] Xiang Yue, Xiaoman Pan, Wenlin Yao, Dian Yu, Dong Yu, and Jianshu Chen. 2022. C-MORE: Pretraining to Answer Open-Domain Questions by Consulting Millions of References. Online. arXiv:2203.08928

[124] ChengXiang Zhai. 2009. *Statistical Language Models for Information Retrieval*. Springer International Publishing. https://doi.org/10.1007/978-3-031-02130-5

[125] Chengxiang Zhai and John Lafferty. 2017. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *ACM SIGIR Forum* 51 (08 2017), 268–276. https://doi.org/10.1145/3130348.3130377

[126] Jingtao Zhan, Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Evaluating Interpolation and Extrapolation Performance of Neural Retrieval Models. Online. arXiv:2204.11447

[127] Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-View Document Representation Learning for Open-Domain Dense Retrieval. Online. arXiv:2203.08372

[128] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval. Online. arXiv:2108.08787

[129] Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. 2022. Towards Best Practices for Training Multilingual Dense Retrieval Models. Online. arXiv:2203.08926

[130] Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An Unsupervised Sentence Embedding Method by Mutual Information Maximization. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:221948991

[131] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. Online. arXiv:2211.14876

[132] Chunting Zhou, Graham Neubig, and Jiatao Gu. 2021. Understanding Knowledge Distillation in Non-autoregressive Machine Translation. Online. arXiv:1911.02727

[133] Mu Zhu. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science* 2 (2004).

[134] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM. https://doi.org/10.1145/3459637.3482243

[135] Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *Comput. Surveys* 38, 2 (jul 2006), 6. https://doi.org/10.1145/1132956.1132959

[136] Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. 1998. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems* 23, 4 (dec 1998), 453–490. https://doi.org/10.1145/296854.277632

A METRICS

The recall@k [133] states the fraction of returned relevant documents in all relevant documents in the corpus

$$\text{Recall@k} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{D_{retr_{q,k}}}{D_{rel_q}} \tag{16}$$

Where:

- Q is the query set over which the recall@k values are averaged
- q is a single query
- $D_{retr_{q,k}}$ is the number of relevant documents for the query q returned at top k positions by the retriever
- D_{rel_q} is the total number of relevant documents for query q.

The precision@k [133] states the fraction of relevant texts in all top-k retrieved texts for the query set Q

$$\text{Precision@k} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{D_{retr_{q,k}}}{k} \tag{17}$$

The average precision (AP_q) [133] averages the Precision values at the position of each correctly returned document for a query.

$$\text{AP}_q = \frac{1}{D_{rel_q}}\sum_{k=1}^{L}\text{Precision@k} \times \mathbb{I}(q,k) \tag{18}$$

Where:

- Precision@k describes the per-query precision at position k
- L is the length of a retrieved list
- $\mathbb{I}(q,k)$ is 1 if the document at position k is relevant for query q, 0 otherwise

The mean average precision (MAP) [133] states the mean precision scores over a set of Queries Q.

$$\text{MAP} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}AP_q \tag{19}$$

The F1 score [133] is the harmonic mean of precision and recall, balancing both metrics to give a single measure of retrieval quality. It is beneficial when there is an uneven class distribution or when both false positives and false negatives are important.

$$\text{F1@k} = \frac{2 \times \text{Precision@k} \times \text{Recall@k}}{\text{Precision@k} + \text{Recall@k}} \tag{20}$$

The discounted cumulative gain (DCG_q) [133] takes the position of the relevant retrieved documents into account and wants relevant texts to be at the top of the list

$$DCG_q@k = \sum_{k=1}^{L}\frac{2^{g_i}-1}{log_2(i+1)'} \tag{21}$$

Where:

- g_i is a graded relevance score for the i-th retrieved text

The normalised discounted cumulative gain (NDCG@k) [133] states the sum of the normalised DCG values at a specific position k.

$$\text{NDCG@k} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{DCG_q@k}{IDCG_q@k} \tag{22}$$

Where:

- $IDCG@k$ is the ideal discounted cumulative gain at a specific rank position k.

The mean reciprocal rank (MRR) [133] forms an average over a set of Queries Q for the reciprocal of the rank of the first retrieved positive document.

$$MRR = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{rank_q} \tag{23}$$

B BASELINE MODELS

The following table summarises the top five baseline model's performance on the test set without fine-tuning.

Table 6: Baseline Results

Metric	German Data	English Data
Mean Precision@10	LLaMA-based (0.711) XLM-R (0.663) XLM-RoBERTa (0.611) LaBSE (0.579) all-MiniLM-L12	XLM-R (0.895) LLaMA-based (0.847) multilingual-MiniLM (0.774) distiluse (0.768) msmarco-distilbert (0.736)
Mean Recall@10	LLaMA-based (0.543) XLM-R (0.528) Multilingual-e5-small (0.518) all-MiniLM-L12 (0.420) BioLORD (0.416)	XLM-R (0.434) LLaMA-based (0.417) multilingual-MiniLM (0.361) Multilingual-e5-small (0.361) all-MiniLM-L12 (0.359)
Mean F1@10	LLaMA-based (0.477) XLM-R (0.440) Multilingual-e5-small (0.391) all-MiniLM-L12 (0.383) XLM-RoBERTa (0.370)	XLM-R (0.547) LLaMA-based (0.523) multilingual-MiniLM (0.463) distiluse (0.459) msmarco-distilbert (0.445)
Mean MRR@10	XLM-R (0.921) LLaMA-based (0.904) all-MiniLM-L12 (0.895) DPR-XM (0.796) LaBSE (0.772)	XLM-R (1.000) LLaMA-based (0.974) DPR-XM (0.965) quora-distilbert (0.954) multilingual-MiniLM (0.947)
Mean Average Precision@10	LLaMA-based (0.821) XLM-R (0.798) all-MiniLM-L12 (0.758) XLM-RoBERTa (0.720) DPR-XM (0.720)	XLM-R (0.903) LLaMA-based (0.870) quora-distilbert (0.854) multilingual-MiniLM (0.828) DPR-XM (0.824)
Mean NDCG@10	LLaMA-based (0.562) XLM-R (0.539) all-MiniLM-L12 (0.462) Multilingual-e5-small (0.455) XLM-RoBERTa (0.441)	LLaMA-based (0.515) XLM-R (0.513) multilingual-MiniLM (0.453) distiluse (0.450) msmarco-distilbert (0.450)

C FURTHER IMPLEMENTATION DETAILS

C.1 Prompt-Based Query Generation

Our implementation of algorithm 2 includes a caching mechanism and the option to send the documents to the GPT-4o instance in parallel to speed up the process.

C.2 Negative Selection

We reimplemented the NegativeMiner since we load the corpus query and qrel data differently for the prompt-based and keyword-generated queries. The original NegativeMiner implements the BEIR data loader directly; we pass the query, corpus, and qrel files instead.

D HISTORY OF TEXTUAL INFORMATION RETRIEVAL

The textual information retrieval task was theorised as early as 1950 by T. Joyce and R. M. Needham [47]. They asked how texts can be indexed and picked representative terms to retrieve relevant information. Following that idea, the "bag-of-words" assumption led to a vector space model that encoded the query and the document in sparse term-based vectors [97, 99]. Many different approaches to constructing these sparse vectors exist. A first approach, called tf-idf, allows for a relevance measure based on the text vector's and query vector's lexical similarity [4, 93, 98]. As a second approach, inverted indexing utilises the same representation idea as tf-idf but makes text retrieval more efficient by organising documents in term-oriented posting lists [135, 136]. Probabilistic relevance frameworks are a third approach to solving the text retrieval task and modelling the relevance of documents. They enable a more in-depth understanding of the retrieval mechanisms. One example of such a framework is BM25 [94, 95], but multiple more (statistical) language modelling approaches exist [124].

The rise of machine learning approaches led to the use of learning to rank algorithms for text retrieval tasks [56, 63]. These algorithms apply supervised machine learning models to rank the relevance of retrieved documents but require human feature engineering.

The availability of more computational power and the application of stochastic gradient descent for neural networks resulted in neural information retrieval, a deep learning approach to text retrieval tasks. This approach no longer required hand-designed feature engineering [33, 34, 44, 79]. Neural information retrieval models map the query and the text corpus into low-dimensional vectors in latent representation space. Thus, they encode the textual input in dense vectors. This process is also called embedding. The similarity between the query and the document vector estimates the relevance of a text for a query. Thus, the relevance of information is no longer estimated by lexical but by semantic similarity. Contrary to this approach, the sparse vectors constructed by classical vector space models encode explicit term dimensions. In contrast, the dense vectors constructed by neural IR models encode the latent semantic characteristics of language. The so-called transformer architecture was the subsequent historical development in the textual information retrieval task after neural IR. Nowadays, transformers are the basis for most NLP tasks. They were initially proposed to model any sequence data by utilising the self-attention mechanism through which every token attends to every other token in the sequence [20, 39, 112]. Their two significant advantages are that they can be trained in parallel and scaled up quickly. The use of transformers for NLP tasks, as well as the availability of large-scale labelled retrieval datasets like MS MARCO[8], gave rise to pre-trained large language models [12, 24, 25, 66]. The pre-trained LLMs use different self-supervised loss functions and are trained on large-scale general document corpora. Fine-tuning the models enables their transfer to downstream tasks [62, 131]. They facilitated a broader representation of semantics and language in general. The most common pre-trained language model is called BERT [24]. It employs a deep bi-directional architecture and word masking to improve text encoding into dense vectors. BERT can encode the general English language. The recent developments after BERT can be categorised into three different research fields. Firstly, different pre-training approaches are explored [66]. Secondly, the bi-directional representation is refined [72], and thirdly, a compressed and thus more lightweight version of BERT was developed, called distilBERT [100]. Nowadays, pre-trained dense retrievers are the "gold standard" for solving textual IR tasks due to their astonishing representation of documents and ability to answer more complex queries [131]. However, most pre-trained dense retrievers were trained on English data containing general, informal knowledge. Transforming them to other languages or specialised domains requires further fine-tuning of labelled data in the desired domain and language. Such data, as mentioned in Section 1, is rarely available.

E INFORMATION RETRIEVAL SYSTEMS

Most information retrieval systems consist of pipelines that combine multiple steps as visualised in Figure 7.

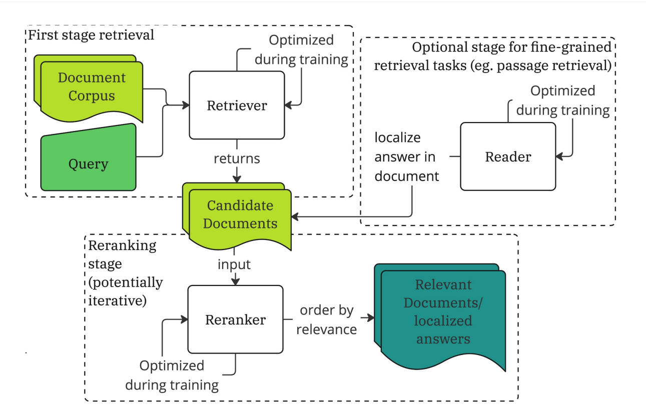
First stage retrieval
Optimized during training
Document Corpus
Retriever
Query
returns
Optional stage for fine-grained retrieval tasks (eg. passage retrieval)
Optimized during training
localize answer in document
Reader
Candidate Documents
Reranking stage (potentially iterative)
input
Reranker
order by relevance
Relevant Documents/ localized answers
Optimized during training

Figure 7: Information Retrieval Systems

The first stage performs the retrieval. Thus, the model selects several candidate documents relevant to the user query. The model which implements this stage is called the retriever. During the second, the re-ranking stage, the selected candidates are ordered by importance. The model which implements this stage is called the reranker. Textual IR systems can utilise one to many re-ranking stages to refine the results. A model called the reader performs the third stage of textual information retrieval. We only add it if the information retrieval task is more fine-grained than document-level retrieval. The reader analyses the documents the retriever returned and localises the response to the query [131].

E.1 Dense Retrievers

In semantic search systems, the retriever, reader, and re-ranker are dense retrievers. However, dense retrievers can be divided into two mainstream architecture types: cross- and bi-encoders. We introduce both in the following subsection.

E.2 Bi-encoders, Cross-encoders or Both?

First, we introduce bi-encoders and, afterwards, cross-encoders. However, Figure 8 allows a direct, visual comparison of both architecture types. We finalise this section by analysing the advantages and disadvantages of both architectures.

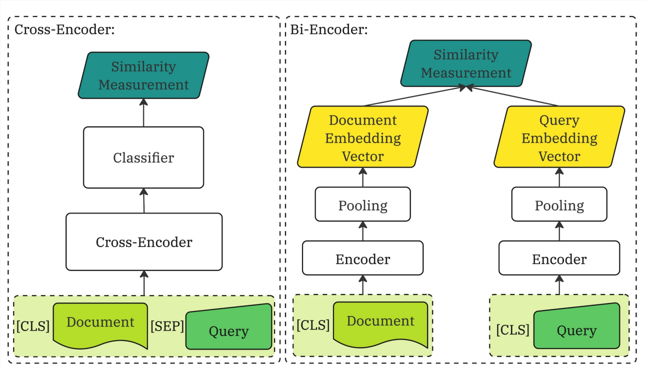

Figure 8: Dense Retriever Architecture Types

As shown in Figure 8, bi-encoders are based on a two-tower architecture, utilising two encoders. They perform the self-attention mechanism for query and document separately, mapping each to the same dense vector space and then using a similarity measurement to measure the distance between both [44, 113, 131]. Their two-tower architecture makes bi-encoders flexible since document and query encoder architecture can differ. It resulted in two different bi-encoder structures. The first is called single-representation bi-encoder. It embeds the query and text separately using two different pre-trained large language models [48, 84, 118]. We place the special [CLS] token at the beginning of the query, and the text and their encodings represent their semantic meaning. The relevance score for a query text combination is then computed through some similarity function using the [CLS] token embeddings. The downside of the single-representation bi-encoder structure is that it does not accurately capture the semantic information of query and text. The second bi-encoder structure is called multi-representation bi-encoder. It aims at enhancing the semantic information stored in the embeddings by combining multiple query and document embeddings to measure their similarity from different "semantic viewpoints". These contextualised embeddings are learned and stored during training and indexing. This second bi-encoder structure improves the retrieval performance compared to the single-representation structure. However, storing the multiple generated views makes their index large, leading to higher computational and memory costs. Examples for multi-representation bi-encoders are poly-encoder [45], ME-BERT [68], ColBERT [49], ColBERTer [41], MVR [127] and MADRM [50].

In contrast to bi-encoders, cross-encoders receive the concatenated query and text pair, distinguished by a separation token [SEP], as input as shown in Figure 8[83]. They utilise a cross-attention mechanism to compute the interaction between any two tokens in the input and encode every token in vector space. Thus, cross-encoders enable the tokens to interact across queries and text [33, 117]. Using the learned representations, we can then compute match representations for the query-text pair. Primarily, the encoding of the [CLS] token is used for this semantic matching [113, 131]. Nevertheless, we can also average over all token embeddings to calculate the similarity measurement [88]. The cross-encoder's output is a fine-grained relevance score for the query text pair, and a higher score indicates higher relevance of the text, given the query.

Utilising a bi-encoder or a cross-encoder has different up- and down-sides. Their two-tower architecture makes bi-encoders computationally more efficient than cross-encoders since approximate nearest neighbour search enables fast recall of large-scale vectors, and only the query, but not the textual information, must be encoded during query time. Since cross-encoders calculate relevance scores for every possible query and text pair, end-to-end information retrieval is not possible, as they do not create independent representations of query and text. Thus, the encoding must be recomputed every time for every query text pair [113, 131]. However, bi-encoders need more training data than cross-encoders since they independently map the documents to vector space. They also reach lower retrieval accuracy.

Due to these pros and cons, bi- and cross-encoders are suited for different tasks in information retrieval systems [131]. Bi-encoders are used in first-stage retrieval to fetch the candidate documents, while Cross-encoders are adapted as re-rankers or readers. In general, cross-encoders perform better if applied to zero-shot retrieval tasks on out-of-domain data, and multiple methods of domain adaptation dense retriever in low-resource setups rely on cross-encoder architectures [113].