UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Thiago Soares Laitz

InRanker:
Ranqueadores Destilados para Recuperação de Informação Zero-shot

InRanker:
Distilled Rankers for Zero-shot Information Retrieval

Campinas
2025

Thiago Soares Laitz

InRanker: Distilled Rankers for Zero-shot Information Retrieval

InRanker: Ranqueadores Destilados para Recuperação de Informação Zero-shot

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, the area of Computer Engineering.

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na área de Engenharia de Computação.

Orientador: Roberto de Alencar Lotufo
Coorientador: Rodrigo Frassetto Nogueira

Este trabalho corresponde à versão final da dissertação defendida pelo aluno Thiago Soares Laitz, e orientada pelo Prof. Dr. Roberto de Alencar Lotufo.

Campinas
2025

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

L146i

Laitz, Thiago Soares, 1998-
InRanker : distilled rankers for zero-shot information retrieval / Thiago Soares Laitz. – Campinas, SP : [s.n.], 2025.

Orientador: Roberto de Alencar Lotufo.
Coorientador: Rodrigo Frassetto Nogueira.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de linguagem natural. 2. Aprendizado profundo. 3. Inteligência artificial. 4. Recuperação de informação. I. Lotufo, Roberto de Alencar, 1955-. II. Nogueira, Rodrigo Frassetto, 1986-. III. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações complementares

Título em outro idioma: InRanker : ranqueadores destilados para recuperação de informação zero-shot
Palavras-chave em inglês:
Natural language processing
Deep learning
Artificial intelligence
Information retrieval
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Roberto de Alencar Lotufo [Orientador]
Levy Boccato
Wagner Meira Junior
Data de defesa: 07-01-2025
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-7205-2094
- Currículo Lattes do autor: http://lattes.cnpq.br/3296408036306084

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Thiago Soares Laitz RA: 224898
Data da Defesa: 07 de janeiro de 2025
Título da Dissertação:
"InRanker: Ranqueadores Destilados para Recuperação de Informação Zero-shot"
"InRanker: Distilled Rankers for Zero-shot Information Retrieval"

Prof. Dr. Roberto de Alencar Lotufo (Presidente)
Prof. Dr. Wagner Meira Junior
Prof. Dr. Levy Boccato

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

"In the pursuit of knowledge, research is not just the discovery of new information, but the continuous refinement of understanding that shapes the future."

"Na busca pelo conhecimento, a pesquisa não é apenas a descoberta de novas informações, mas um refinamento contínuo do entendimento que molda o futuro."

Thiago Soares Laitz

Resumo

Desde o surgimento dos modelos de linguagem baseados na arquitetura *transformers*, seu desempenho tem superado significativamente o de modelos não neurais em tarefas de NLP (do inglês, *natural language processing*) e IR (do inglês, *information retrieval*). Diante desse cenário, os sistemas de recuperação de informação estado-da-arte têm sido fundamentados nessas arquiteturas. Contudo, o desempenho desses sistemas é diretamente proporcional ao número de parâmetros dos modelos, o que resulta em um aumento substancial dos custos e da latência, prejudicando a implementação desses modelos em ambientes que demandam respostas em tempo real e/ou possuem recursos limitados. Modelos como o T5, um modelo transformer sequência-para-sequência (seq2seq), possui versões que extrapolam 11 bilhões de parâmetros, demandando *hardware* com alto poder computacional, como múltiplas GPUs (*graphic processing units*) ou TPUs (*tensor processing units*), para operar em cenários de baixa latência.

Este trabalho de mestrado visa explorar novas estratégias de transferência de conhecimento em recuperação de informação com foco no desempenho fora do domínio de treinamento. O objetivo é reduzir o tamanho dos modelos utilizados em sistemas de recuperação de informação, sem prejudicar a capacidade de generalização para dados fora do domínio de treinamento do modelo (cenários de *zero-shot*). Assim, foi criado o InRanker, um modelo estudante derivado de modelos professores maiores, que emprega uma abordagem de destilação em duas etapas. Na primeira etapa, o modelo estudante é treinado de forma supervisionada a partir de uma base de dados rotulados de IR suficientemente grande e diversa, usando rótulos suaves produzidos pelo modelo professor. Na segunda etapa, são criadas perguntas sintéticas a partir de grandes modelos de linguagem (LLMs, do inglês *large language models*), gerando dados que imitam o domínio-alvo e ampliam o escopo de conhecimento do modelo estudante para situações fora do domínio original de treinamento. Com isso, busca-se diminuir o tamanho do modelo e assegurar que ele mantenha alta eficácia em cenários *zero-shot*, semelhante ao comportamento encontrado em grandes modelos.

A metodologia proposta foi avaliada em conjuntos de dados de recuperação de informação, como o BEIR, que abrange uma grande variedade de domínios textuais. Os resultados indicam que modelos substancialmente menores, como o monoT5-60M e o monoT5-220M, conseguem atingir níveis de desempenho comparáveis ao de seu modelo professor (monoT5-3B), mesmo sendo 50 e 13 vezes menores, respectivamente. Isso evidencia que a abordagem de destilação sugerida não só possibilita uma redução de custos e de exigências computacionais, mas também preserva a capacidade de generalização, aspecto essencial para a aplicação em contextos reais e diversos. Adicionalmente, este trabalho também explorou formas de transferir o conhecimento em cenários multilíngues, nos quais o professor foi treinado com dados em inglês e transferiu o conhecimento para o estudante com foco em documentos em português. Os modelos e códigos utilizados estão disponíveis em `https://github.com/unicamp-dl/InRanker`.

Palavras-chave: Processamento de Linguagem Natural; Aprendizado Profundo; Modelos de Linguagem; Recuperação de Informação

Abstract

Since the emergence of neural language models based on the transformer architecture, the semantic and contextual understanding of words has become much more precise compared to their non-neural counterparts on tasks such as NLP (natural language processing) and IR (information retrieval). In this context, state-of-the-art information retrieval systems have been built upon these architectures. However, the efficacy of these systems is directly proportional to the number of parameters in the models, which results in a substantial increase in costs and latency, hindering the implementation of these models in environments that demand real-time responses and/or have limited resources. Models like T5 (a transformer-based sequence-to-sequence model) have versions exceeding 11 billion parameters, requiring specialized hardware, such as GPUs (graphic processing units) or TPUs (tensor processing units), to operate effectively in low-latency scenarios.

This master thesis aims to explore new knowledge transfer strategies in information retrieval, focusing on the out-of-domain effectiveness. The goal is to reduce the size of the models used in information retrieval systems without compromising their ability to generalize to data outside the model's training domain. To this end, InRanker was created, a student model derived from monoT5-3B, which employs a two-step distillation approach. In the first step, the student model is supervisedly trained using a large and diverse labeled IR dataset, using soft labels produced by the teacher model. In the second step, synthetic queries are created using large language models (LLMs), generating data that resembles the target domain and expands the student model's knowledge scope for situations outside the original domain. This approach aims not only to reduce the model size, but also to ensure that it maintains high effectiveness in zero-shot scenarios, similar to the behavior observed in large models.

The proposed methodology was evaluated using information retrieval datasets, such as BEIR, which covers a wide range of textual domains. The results indicate that significantly smaller models, such as monoT5-60M and monoT5-220M, can achieve performance levels comparable to their teacher model (monoT5-3B), despite being 50 and 13 times smaller, respectively. This demonstrates that the suggested distillation approach not only enables cost and computational requirement reductions, but also preserves generalization capacity, an essential aspect for applications in real-world and diverse contexts. Additionally, this work also explored ways to transfer knowledge in multilingual scenarios, where the teacher was trained with data in English and transferred knowledge to the student model focusing on documents in Portuguese. The models and codes used are available at `https://github.com/unicamp-dl/InRanker`.

Keywords: Natural Language Processing; Deep Learning; Language Models; Information Retrieval

List of Figures

List of Tables

Contents

1 Introduction

As the volume of data continues to grow exponentially [1], retrieving relevant information has become essential for a variety of applications, from search engines to decision-making systems. The challenge lies in efficiently processing documents while accurately predicting the user's intent behind a query. Information Retrieval (IR) refers to the computational process of identifying and extracting relevant information from those source of data, in response to a specific query. This task involves not only the retrieval of documents but also understanding and interpreting the semantics of both the query and the content within the documents, making the process particularly complex. Queries are typically concise, and the system must infer the user's intent and locate documents that are relevant to the query. Additionally, the volume of data in modern digital repositories adds to the challenge, as processing these large datasets requires efficient algorithms and considerable computational resources.

Over the last few years, new approaches have been developed to handle the key challenges in IR. Traditional models, such as boolean search and probabilistic models, have been foundational, but more recent advancements in machine learning, natural language processing, and deep learning have led to newer techniques that are capable of better handling linguistic nuances, context, and the high dimensionality of modern data.

These techniques rely on neural networks, usually based on the transformer architecture, which may reach billions of parameters. In fact, it is well known that the effectiveness of IR pipelines increases with the number of parameters [3, 26, 28, 29, 33]. For instance, multi-billion parameter rankers and dense models achieve top positions on leaderboards of IR benchmarks and competitions [11, 12, 13]. These large models leverage increased representation capacity, enabling them to encode features that might elude smaller models. However, deploying these large models is not without its challenges. The computational overheads are substantial, often requiring specialized hardware such as GPUs or TPUs to operate in latency-critical applications. The high cost is directly related to the large number of parameters that these models contain, as they require hardware with high memory and computational capacity. In a production environment, this means higher operating costs and reduced scalability.

To address these challenges, there have been efforts to create more efficient models without significantly reducing effectiveness. One such approach is model distillation [18]. Distilled models, such as MiniLM [37], use a teacher or an ensemble of larger models to transfer knowledge to a smaller student model. Rosa et al. [35] show that MiniLM surpassed the zero-shot effectiveness of monoT5-base in IR tasks despite being an order of magnitude smaller in size. This has shown that knowledge transfer via model distillation is not only feasible but also effective. However, most distillation techniques have been geared towards optimizing effectiveness on specific benchmark tasks and do not focus on out-of-domain effectiveness. Rosa et al. also show that while smaller models are capable of achieving high in-domain results, similar to their larger counterparts, the disparity in effectiveness becomes evident in out-of-domain scenarios.

Usually, training a retrieval model requires human-annotated hard labels informing which passage is relevant for each query. However, with the advance of Large Language Models (LLMs), it has become possible to generate synthetic queries for passages, providing a feasible approach for data augmentation [3, 20, 5, 30, 2]. Our work introduces a method for the generation of synthetic data specifically designed for distilling rankers that increases their out-of-domain effectiveness. We present InRanker, a distilled model derived from monoT5-3B [29], that uses the predictions of the teacher directly with both synthetic, generated from an out-of-domain corpus, and real query-document pairs. Effectively, this approach converts any corpus to be in-domain, since the model will be trained using queries from the target domain. As a result, this approach leads to reduced model sizes while maintaining improved out-of-domain effectiveness, as presented in Figure 12. The research conducted during the master's program and documented in this dissertation also resulted in a scientific article made available as a preprint on arXiv [22] and published on Bracis 2024.

This document is structured as follows: it begins with an introduction Chapter 1 that provides background information, including the motivation, evaluation metrics, datasets, and main contributions. This is followed by a Chapter on related work 2, a Chapter detailing the proposed pipeline 3, a Chapter presenting the results 4, and finally, a Chapter discussing the conclusions and limitations of this study 5.

1.1 Background

In this section, we introduce the main concepts of information retrieval, evaluation metrics, and explain different approaches commonly used to this task. We also present the monoT5 model, which is the main focus of this work.

In addition to the development of retrieval algorithms, the research community has created metrics to assess the performance of IR systems. Metrics such as precision, recall and more complex measures like mean average precision (MAP) or normalized discounted cumulative gain (nDCG) are used to quantify the accuracy and relevance of candidate documents to the user's query. In the following sections we present in detail the main evaluation metrics, ranking approaches and main evaluation datasets.

1.1.1 Evaluation Metrics

There are many metrics used to evaluate the performance of IR systems. Some of the most used metrics are:

- **Recall**: the fraction of relevant documents that have been retrieved over the total number of relevant documents in the corpus. It is calculated as the number of relevant documents retrieved divided by the total number of relevant documents available, where d is the set of documents retrieved by the system and D the set of all relevant documents in the corpus.

$$\text{Recall} = \frac{|d \cap D|}{|D|} \tag{1}$$

  This metric is crucial for understanding how effectively a search system is able to cover the relevant documents in the corpus, indicating the system's ability to not miss out relevant information.

- **Precision**: the fraction of relevant documents among the retrieved documents. It is calculated as the number of relevant documents retrieved divided by the total number of documents retrieved, and expresses how reliable is the model pointing out documents that actually are relevant within the retrieved documents:

$$\text{Precision} = \frac{|d \cap D|}{|d|} \tag{2}$$

- **Mean Average Precision (MAP)**: it corresponds to the arithmetic mean of the Average Precision (AP) values across all queries. For each query q, the Average Precision is calculated by determining the precision at each rank k (cutoff point) where a relevant document is retrieved, and then averaging these precision values over the total number of relevant documents for that query D_q. The MAP is the mean of these AP values across all queries in the query set Q. In this context, Q represents the set of all queries, $|Q|$ is the total number of queries, D_q is the set of relevant documents for query q, n_q is the total number of retrieved documents for query q, $P_q(k)$ is the precision at rank k for query q, and $\text{rel}_q(k)$ is an indicator function that is 1 if the document at rank k for query q is relevant, and 0 otherwise. The equation for MAP is:

$$\text{MAP} = \frac{1}{|Q|} \sum_{q \in Q} \left(\frac{1}{|D_q|} \sum_{k=1}^{n_q} P_q(k) \cdot \text{rel}_q(k) \right) \tag{3}$$

- **Mean Reciprocal Rank (MRR)**: a measure of a system's ability to return highly ranked relevant documents, emphasizing the importance of having relevant documents appearing as early as possible in the search results. It is calculated as the average of the reciprocal ranks of the first relevant document for each query in a set of queries. The reciprocal rank is the inverse of the rank at which the first relevant document is retrieved. The metric is defined as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}(q)}, \tag{4}$$

where rank(q) is the rank of the first relevant document for query q.

- **Normalized Discounted Cumulative Gain (nDCG)**: a metric for evaluating the quality of a set of search results, especially when the relevance of each document is not binary but graded. It measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom, with the gain of each document discounted at lower ranks. This metric is particularly useful for situations where the most relevant documents appearing earlier in the search results are more beneficial to the user. The nDCG is normalized against the ideal order of documents (the maximum possible DCG), to ensure the score is within a range from 0 to 1, making it easier to compare between different sets of results. The formula for nDCG, considering a list of retrieved documents up to a particular rank K, is:

  $$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{5}$$

  where $DCG@K$ (Discounted Cumulative Gain) is calculated as:

  $$DCG@K = \sum_{k=1}^{K} \frac{2^{rel_k} - 1}{\log_2(k+1)} \tag{6}$$

  with rel_k being the graded relevance of the document at position k.

  $IDCG@K$ is the ideal DCG at rank K, representing the maximum possible DCG given the set of document relevance grades. This is calculated by sorting documents by their relevance in descending order and then applying the DCG formula. The division of $DCG@K$ by $IDCG@K$ ensures that the score is normalized, allowing for meaningful comparisons across different search result sets or queries by accounting for the varying levels of document relevance.

1.1.2 Information Retrieval Datasets

A commonly used dataset for training and evaluating IR systems is the MS MARCO [27], which is a large-scale dataset that contains 6980 queries, 8.84M documents, and the relevance judgments for each query-document pair (there are 400 thousands in this dataset). The dataset is widely used in the community and has been employed in many competitions and challenges.

Also, there are benchmarks composed of different datasets such as BEIR (Benchmark for Information Retrieval) [36], which contains a broad variety of text domains for testing the effectiveness of IR systems. The benchmark is composed of nine retrieval tasks: fact-checking (FEVER, Climate-FEVER, SciFact); question answering (NQ, HotpotQA, FiQA-2018); biomedical IR (TREC-COVID, BioASQ, NFCorpus); news retrieval (TREC-News, Robust04); argument retrieval (Touché-2020, ArguAna); duplicate question retrieval (Quora, CQADupStack); citation prediction (SciDocs); tweet retrieval (Signal-1M); and entity retrieval (DBpedia).

1.1.3 Ranking Approaches

In Information Retrieval (IR), an essential aspect of how systems retrieve and rank relevant documents lies in the way text is represented. Two primary approaches to this representation are sparse and dense models.

Sparse Models

Sparse models, such as Term Frequency-Inverse Document Frequency (TF-IDF) and BM25, use high-dimensional vectors where each dimension corresponds to a specific term in the vocabulary. These vectors are typically sparse, meaning they contain many zero values because most documents include only a small subset of possible terms. This approach has been foundational in IR due to its interpretability and simplicity. However, traditional sparse models may struggle with capturing deeper semantic relationships between words.

- **Computational Efficiency:** Since most positions in these vectors are zeros, sparse models can be stored and processed efficiently. Algorithms can skip over the zeros, focusing only on the non-zero values, which speeds up calculations.

- **Interpretability:** Sparse models are easy to understand. For instance, if a document scores high on a particular term, you can directly see that the presence of that term is important for retrieval. This transparency allows researchers to analyze and fine-tune the retrieval process by understanding which terms contribute most to the ranking.

- **Inability to Capture Synonyms and Polysemy:** Traditional sparse models struggle with understanding relationships between different words that have similar meanings (synonyms) or words that have multiple meanings (polysemy). For example:

- A query for “automobile” might not retrieve a document that uses the word “car” instead, even though they mean the same thing.
- Similarly, the word “bank” could refer to a financial institution or the side of a river. Sparse models may not effectively disambiguate these meanings based on the context.

A commonly used non-neural sparse algorithm is **BM25**, which is a ranking function used by search engines to rank documents according to their relevance to a given query, based on a probabilistic model. It improves upon the TF-IDF approach by incorporating document length and query term frequency to adjust the weighting term. In BM25, each term in the query is assigned a weight based on its frequency in the document, its presence in the document collection, and the length of the document. This method acknowledges that longer documents may have more occurrences of a term purely by chance, and adjusts for this by normalizing term frequencies against document length. Furthermore, BM25 introduces the concept of saturation, which limits the benefit of additional occurrences of a term in a document, reflecting the intuition that after a certain point, more occurrences of a word do not make the document more relevant. BM25 is widely regarded for its effectiveness and is a standard benchmark in information retrieval tasks. It is defined by the following formula:

$$\text{BM25}(D,Q) = \sum_{i=1}^{n} \text{IDF}(t_i) \times \frac{f(t_i,D) \times (k_1+1)}{f(t_i,D) + k_1 \times (1 - b + b \times \frac{|D|}{\text{avgdl}})} \tag{7}$$

where D is the document, Q is the query, n is the number of terms in the query, $f(t_i, D)$ is the frequency of term t_i in document D, k_1 and b are free parameters, and avgdl is the average document length in the collection. The IDF function is defined as:

$$\text{IDF}(t_i) = \ln\left(\frac{N - n(t_i) + 0.5}{n(t_i) + 0.5} + 1\right) \tag{8}$$

where N is the total number of documents in the collection and $n(t_i)$ is the number of documents containing term t_i.

Dense Models

Dense models, on the other hand, are a more recent development, emerging from advances in deep learning, especially with transformer-based architectures. These models represent documents and queries as dense vectors that capture the semantic meaning in a continuous vector space. Unlike sparse models, where each dimension is directly tied

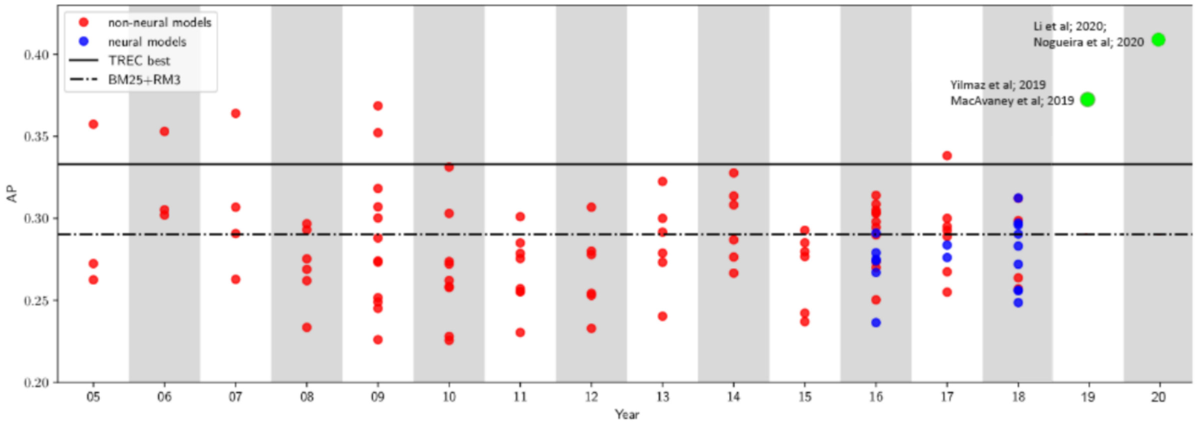

Figure 1: Evolution of state-of-the-art average precision on Robust-04 of neural and non-neural approaches. Adapted from [38].

to a specific term, dense models learn abstract features that capture semantic nuances and contextual dependencies. This allows dense models to perform well in tasks requiring deep understanding of the text's underlying meaning, particularly in complex queries or languages with rich morphological structures. However, dense models come with their own set of challenges, such as reduced interpretability and a requirement for significant computational resources during training, document encoding and retrieval.

Improving Sparse and Dense Models with Neural Models

Both sparse and dense models can be significantly enhanced by leveraging neural network-based approaches, which allow for more sophisticated and effective text representations. Neural models, particularly those based on deep learning, offer the ability to capture complex semantic relationships and contextual nuances that traditional methods may miss. Among these neural models, two prominent architectures are bi-encoders and cross-encoders, each with its own strengths and applications. However, these approaches had only been able to improve upon non-neural models with the release of models such as BERT and T5, as illustrated in Figure 1, which shows the average precision on Robust-04.

Bi-Encoders: Bi-encoders are a class of models used for IR that encode the query and the document separately and then compute the similarity between the two encodings. The most common approach is to use a transformer or BERT-based model to encode the query and the document, and then compute the similarity between the two encodings using a dot product or a cosine similarity. Bi-encoders are simple and efficient, but they have limitations, such as not being able to capture the interaction between the query and the document. Figure 2 illustrates the approach used by Colbert (a bi-encoder retriever) to reduce retrieval latency, in which the query and text interact only after their representations are computed (late-interaction). This approach allows for the document representations to be computed beforehand, since the corpus being searched is already

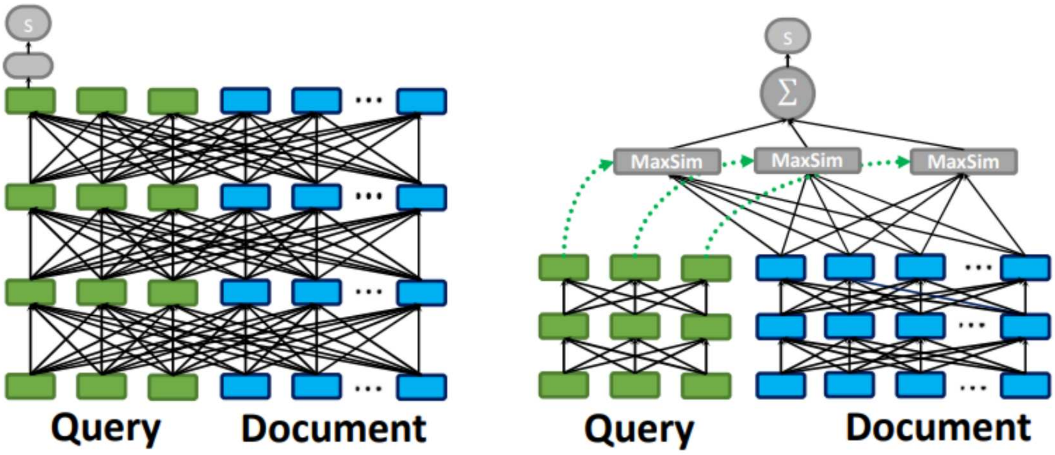

Figure 2: All-to-all interaction on the left versus late interaction (from Colbert) on the right. Adapted from [21].

known.

Cross-Encoders: Cross-encoders, on the other hand, are models that process the query and the document together, allowing the model to capture the interaction between the two texts more effectively. However, cross-encoders tend to be more computationally expensive than bi-encoders, as we cannot pre-compute the encoding of documents beforehand, which means that all computations must be done at inference time.

1.1.4 Rerankers

In addition to sparse and dense retrieval models, rerankers represent another possibility. Unlike sparse and dense models, which generate final representations for documents or queries, rerankers are designed to assign a probability or relevance score to each document-query pair. These scores are then used to sort and rank the search results, refining the output from the initial retrieval stage. Although it is possible to use rerankers to all documents in the corpus, their cross-encoder architecture tends to result in high latency. To mitigate this, rerankers are typically used in a multistage retrieval pipeline. In such pipelines, an initial retrieval model (either sparse or dense) efficiently retrieves a subset of relevant documents, which are then reranked by the reranker, balancing accuracy with computational efficiency. Figure 3 illustrates the architecture of a multi-stage pipeline with multiple rerankers. The core idea is to begin with a fast, but typically less accurate, retriever. At each subsequent stage, the number of candidate documents is reduced and passed to increasingly stronger rerankers, which are usually more effective and have more parameters.

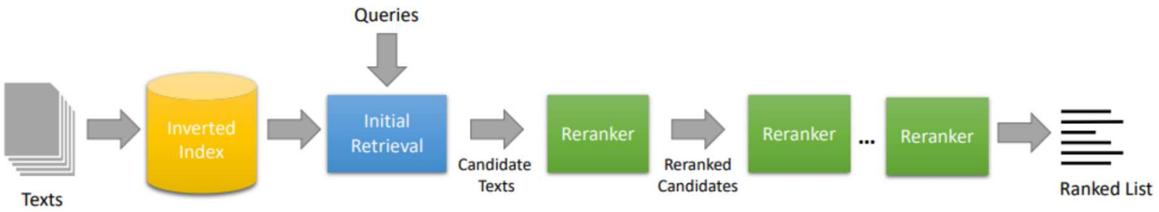

Figure 3: Diagram of a multi-stage ranking architecture. Extracted from [23].

monoT5

A common neural model for IR is a cross-encoder transformer called monoT5. The model proposed an approach to adapt a seq2seq transformer for the IR problem by training it with triples of $\langle \text{query}, \text{document}, \text{relevance} \rangle$ on the MS MARCO dataset using a simple prompt as input:

```
query: <query> document: <document>
```

Although there are different sizes of the model, ranging from 70M to 11B parameters, usually the effectiveness of the model increases with the number of parameters, as shown in Figure 12 in which monoT5 is represented as “hard labels". However, the latency of the model also increases with the number of parameters, which for some applications, such as search engines, increases the response time of the system and the cost of the infrastructure. This work aims to study approaches for effectively distilling the knowledge of a large model into a smaller one, while maintaining the effectiveness of the original model on zero-shot scenarios, i.e, on domains that the model was not trained on.

1.2 Main Contributions

This dissertation makes key contributions to the field of information retrieval and model distillation, specifically in the context of zero-shot generalization for distilled neural rankers, such as:

1. **Introduction of InRanker:** the InRanker models were evaluated on the BEIR benchmark, which comprises 16 datasets from various domains. The experimental results show that InRanker outperforms baseline models of similar sizes and even matches the effectiveness of larger models, especially in out-of-domain retrieval tasks, validating the effectiveness of the proposed distillation technique.

2. **Generalization experiments:** we introduced a methodology to evaluate the generalization capabilities of the models, enabling the simulation of both in-domain and out-of-domain scenarios. This approach did not require any additional annotation, as it relied solely on synthetic queries generated by an LLM.

3. **Portuguese InRanker:** in addition to the English model, this work has demonstrated that it is possible to transfer knowledge from an English teacher model into a Portuguese fine-tuned model, such as PTT5, opening doors for future research in creating new language-tuned rankers.

4. **Open-Source Code and Models:** as part of this research, the codebase and models have been made publicly available to the community, contributing to the transparency and reproducibility of results. This open-source release enables further research in model distillation and efficient neural rankers, particularly in multilingual and resource-constrained environments.

Additionally, a paper based on this master's work was accepted for publication at BRACIS 2024, and the open-source models have been downloaded over more than 22000 times on HuggingFace as of October 2024.

2 Related Work

The research community has been using LLMs in a variety of tasks aimed at increasing the availability of data and improving the effectiveness of existing systems. Magister et al. [24] explored the fact that chain-of-thought reasoning only works for models with tens of billions of parameters and decided to use synthetic chain-of-thought reasoning text generated by PaLM 540B [9] and GPT-3 175B [6] to transfer knowledge to smaller models such as T5 XXL. As a result, the student model improved its task performance across arithmetic, commonsense and symbolic reasoning datasets. Figure 4 shows the process of creating a chain-of-thought dataset for fine-tuning. Fu et al. [16] successfully specialized a FlanT5 [10] student model focusing in multi-step math reasoning using code-davinci-002 as teacher, showing that it is possible to fine-tune smaller models to reasoning tasks that are hard to small models. However, all these works rely on training the student models using synthetic text rather than directly using the soft labels. Furthermore, Muhamed et al. [25] distilled cross-attention scores of a BERT based model for click-through-rate prediction, achieving better results when exposed to contextual features such as tabular data, showing that it is possible to specialize the student models on downstream tasks that use different types of data such as tabular information or natural language content.

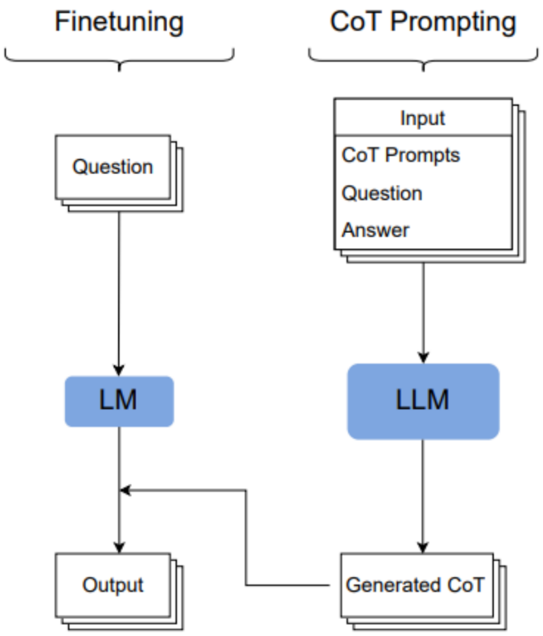

Figure 4: Distillation with Chain-of-thought reasoning. Extracted from [24]

Although it is possible to fine-tune or transfer knowledge using synthetically generated content, Wang et al. [37] distilled the self-attention module of a transformer teacher into a student model with reduced size. Figure 5 shows that during training, a KL-divergence loss is used to make the distribution of the last self-attention layer match between the models. The resultant model has been used in a variety of tasks, including a reranker that achieves competitive results compared to models with the same scale of parameters [35].

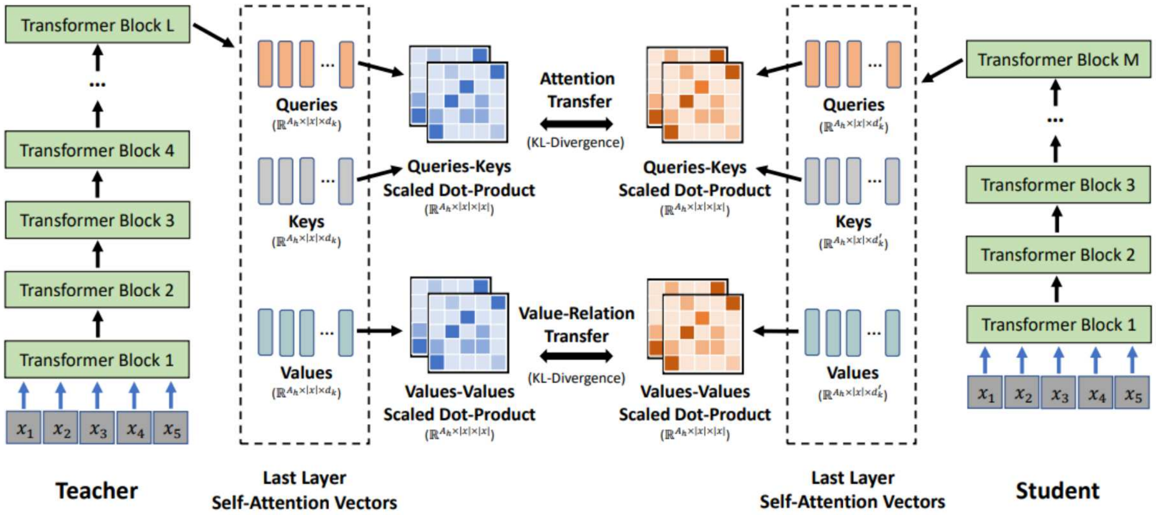
Transformer Block L
...
Transformer Block 4
Transformer Block 3
Transformer Block 2
Transformer Block 1
x_1 x_2 x_3 x_4 x_5
Teacher
Queries
$(\mathbb{R}^{A_h \times |x| \times d_k})$
Keys
$(\mathbb{R}^{A_h \times |x| \times d_k})$
Values
$(\mathbb{R}^{A_h \times |x| \times d_k})$
Last Layer
Self-Attention Vectors
Queries-Keys
Scaled Dot-Product
$(\mathbb{R}^{A_h \times |x| \times |x|})$
Values-Values
Scaled Dot-Product
$(\mathbb{R}^{A_h \times |x| \times |x|})$
Attention
Transfer
(KL-Divergence)
Value-Relation
Transfer
(KL-Divergence)
Queries-Keys
Scaled Dot-Product
$(\mathbb{R}^{A_h \times |x| \times |x|})$
Values-Values
Scaled Dot-Product
$(\mathbb{R}^{A_h \times |x| \times |x|})$
Queries
$(\mathbb{R}^{A_h \times |x| \times d_k'})$
Keys
$(\mathbb{R}^{A_h \times |x| \times d_k'})$
Values
$(\mathbb{R}^{A_h \times |x| \times d_k'})$
Last Layer
Self-Attention Vectors
Transformer Block M
...
Transformer Block 3
Transformer Block 2
Transformer Block 1
x_1 x_2 x_3 x_4 x_5
Student

Figure 5: Illustration of self-attention distillation of MiniLM. Extracted from [37].

Previous studies have also explored training a student from soft labels produced by a teacher, i.e., outputs that are not binary. For instance, Hofstätter et al. [19] proposed a cross-architecture knowledge distillation approach using the MarginMSE loss, where the primary goal is to minimize the distance between the teacher's and the student's margin outputs. These margin outputs are defined as the difference between the positive class logit and the negative class logit. The loss is formalized in Equation (9), where M_t represents the teacher's output, M_s represents the student's output, Q is the given query, and P refers to the passages that could either be relevant (positive) or non-relevant (negative) to the query. Figure 6 shows the diagram of knowledge distillation using the MarginMSE loss.

$$\text{MarginMSE} = \text{MSE}(M_s(Q, P+) - M_s(Q, P-), M_t(Q, P+) - M_t(Q, P-)) \tag{9}$$

Similarly, Formal et al. [15] used the MarginMSE loss to distill knowledge for sparse neural models, such as SPLADE. In this approach, the model is capable of generating a sparse representation of documents and queries considering the text context and synonyms. This allows the distilled model to capture more nuanced meanings within the text while maintaining sparsity in its representations, which is essential for efficient retrieval in large-scale information retrieval systems. Finally, Hashemi et al. [17] proposed a method for generating synthetic data for domain adaptation of dense passage retrievers. This approach involves creating new queries and a target collection, along with pseudo-labels

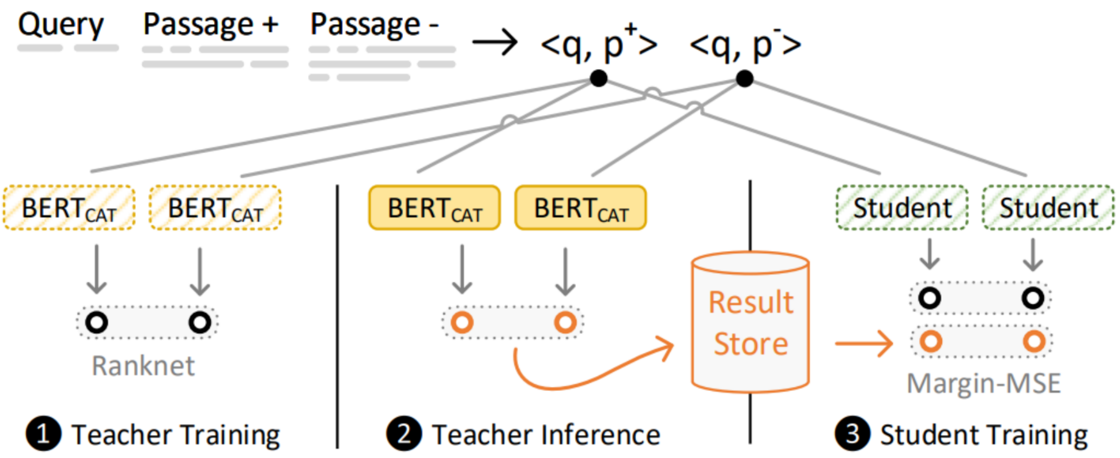

Figure 6: Diagram of knowledge distillation process using the MarginMSE loss. Extracted from [19].

extracted using a BERT cross-encoder. However, they did not evaluate the model's effectiveness on datasets to which it was not domain-adapted. The existing research has mainly focused on in-domain evaluation, where the goal has been to increase the effectiveness of the student model on test datasets whose domain is similar to the datasets it was trained on. Our study also focuses on the robustness of the student and its ability to perform well even in out-of-domain scenarios, similar to the abilities of the larger teacher model.

Finally, InPars [3] presents a methodology for automatically generating synthetic samples to train IR models, such as a reranker. This approach uses an LLM to generate synthetic queries for a given passage, followed by probabilities that are used to filter the top K query-document pairs, which are then used as positive samples during training. InPars-v2 [20] follows the same procedure but adds a reranker into the pipeline to better select training pairs. Figure 7 illustrates the fine-tuning process. The authors claim that models trained on synthetic datasets generated by InPars were able to outperform baselines like BM25. Furthermore, models trained with a combination of synthetic and supervised datasets showed improved zero-shot effectiveness. However, it requires training a separate model for each target domain. Conversely, the proposed approach (InRanker) shows better generalization when fine-tuned on all 16 selected BEIR datasets simultaneously.

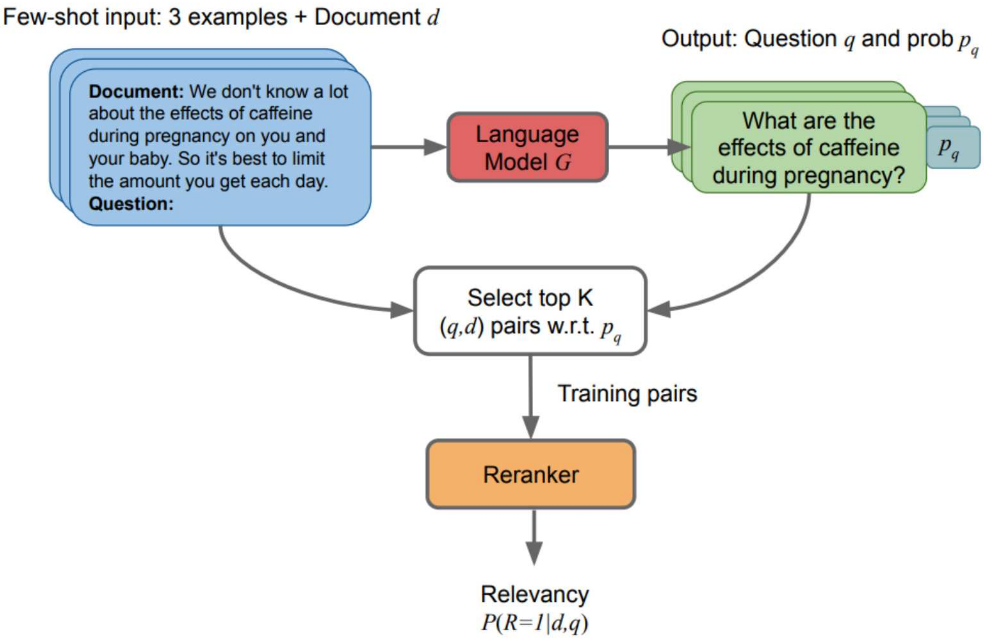
Few-shot input: 3 examples + Document d
Document: We don't know a lot about the effects of caffeine during pregnancy on you and your baby. So it's best to limit the amount you get each day.
Question:
Language Model G
Output: Question q and prob p_q
What are the effects of caffeine during pregnancy?
p_q
Select top K (q,d) pairs w.r.t. p_q
Training pairs
Reranker
Relevancy
$P(R=1|d,q)$

Figure 7: InPars synthetic dataset generation pipeline. Extracted from [3].

3 Proposed Method

3.1 Training

Our proposed method consists of two key phases of distillation, each designed with specific objectives to maximize the model's zero-shot effectiveness. The first phase uses real-world data to familiarize the student model with the ranking task, while the second phase uses synthetic data designed to improve zero-shot generalization and improve the model's effectiveness on specific datasets. The datasets used to distill InRanker consists of {query, passage, logits} triplets, where the logits (soft labels) come from a teacher model that has been trained for the relevance task. For the first stage, we chose to use query-document pairs from the MS MARCO [27] dataset, given their variety, the large number of annotated pairs, and its demonstrated effectiveness in enhancing retrieval effectiveness [34]. Figure 8 shows an example of query and passages from MS MARCO.

Query: at what age do kids start to hold memories?

Relevant Passage: Childhood amnesia, also called infantile amnesia, is the inability of adults to retrieve episodic memories before the age of 2 to 4 years, as well as the period before age 10 of which adults retain fewer memories than might otherwise be expected given the passage of time.

Non-relevant Passage: In an effort to better understand how children form memories, the researchers asked 140 kids between the ages of 4 and 13 to describe their earliest memories and then asked them to do the same thing two years later.

Figure 8: Sample from the MS MARCO dataset. A query and a relevant and non-relevant passage.

Next, we source synthetic queries from InPars [3], which used an LLM to create queries for the datasets in BEIR in a few-shot manner. The synthetic queries were generated using the prompt shown in Figure 9. The strategy involves providing three examples of queries and passages, followed by a "fake" fourth example, that is actually the one the language model is supposed to complete, after which it is extracted and saved as a training sample.

After the datasets were processed, distilling rerankers involves using the Mean Squared Error (MSE) loss to match the logits of the teacher and the student, as part of a two-phase pipeline illustrated in Figure 10. The first phase consists of two steps: (1) generating the teacher logits given a query and either a positive (relevant) or a negative (non-relevant)

Example 1:
Document: We don't know a lot about the effects of caffeine during pregnancy on you and your baby. So it's best to limit the amount you get each day. If you are pregnant, limit caffeine to 200 milligrams each day. This is about the amount in $1\frac{1}{2}$ 8-ounce cups of coffee or one 12-ounce cup of coffee.
Relevant Query: Is a little caffeine ok during pregnancy?

Example 2:
Document: Passiflora herbertiana. A rare passion fruit native to Australia. Fruits are green-skinned, white fleshed, with an unknown edible rating. Some sources list the fruit as edible, sweet and tasty, while others list the fruits as being bitter and inedible.
Relevant Query: What fruit is native to Australia?

Example 3:
Document: The Canadian Armed Forces. 1 The first large-scale Canadian peacekeeping mission started in Egypt on November 24, 1956. 2 There are approximately 65,000 Regular Force and 25,000 reservist members in the Canadian military. 3 In Canada, August 9 is designated as National Peacekeepers' Day.
Relevant Query: How large is the Canadian military?

Example 4:
Document: (document text)
Relevant Query:

Figure 9: Prompt used by InPars to generate synthetic queries for target domains.

passage, where the negatives are randomly sampled using BM25 on the top-$k = 1000$ candidates, and the positives are sampled from the human-annotated pairs; and (2) training InRanker given the queries and passages as input using the MSE loss to match the student logits to those of the teacher, which remain frozen during training. This approach can be beneficial as it removes the need for making hard decisions about a passage's relevance, i.e. determining a threshold to obtain binary relevance labels, and instead focuses on a soft target objective aimed at aligning the student's perception of relevance with that of the teacher.

The second phase, with a focus on zero-shot effectiveness, involves the same aforementioned. However, instead of employing real queries sourced from a costly human-annotation process, it uses synthetic queries generated by an LLM based on randomly sampled documents from the corpus (InPars). In this scenario, the positive document is the one used to create the query, and the negatives are collected using the same top-k sampling approach as before.

We also perform zero-mean normalization on the teacher logits for each query-document pair, independent of the overall dataset distribution. This approach intends to make the data distribution symmetric for each query-document pair, thereby minimizing the bias that InRanker is required to learn. Formally:

$$\begin{aligned} L'_{\text{true}} &= L_{\text{true}} - \frac{L_{\text{true}} + L_{\text{false}}}{2} \\ L'_{\text{false}} &= L_{\text{false}} - \frac{L_{\text{true}} + L_{\text{false}}}{2}, \end{aligned} \tag{10}$$

with L_{true} and L_{false} denoting the teacher's logits for the relevant and non-relevant classes, respectively, and L' being the normalized values. This results in the following loss for each training example:

$$\mathcal{L}_{\text{MSE}} = ([Y_{\text{true}} - L'_{\text{true}}]^2 + [Y_{\text{false}} - L'_{\text{false}}]^2), \tag{11}$$

with Y_{true} and Y_{false} representing the logits of the student.

Due to the training objective described in Equation (11), the model no longer determines the relevance of passages and instead focuses on replicating the teacher's output, thus eliminating the need for tuning a relevance threshold that would be needed to produce a binary label. With this approach, we can expand the out-of-domain knowledge of distilled models by generating new queries for documents using an LLM and fine-tuning the distilled model using the teacher's logits. In Chapter 4 we show the effectiveness of this approach in improving the student model's effectiveness across 16 datasets of BEIR simultaneously. We present the hyperparameters used for training, the dataset curation, and we discuss variations of the training loss in Section 4.1.

3.2 Evaluation

For the evaluation of the InRanker models, we used 16 datasets from BEIR, which contain a variety of subjects such as science, biomedical, news, and finance. We focused on two evaluation points:

- **Ranking effectiveness**: For this, we fine-tuned the models using synthetic queries from all datasets simultaneously and considered the average nDCG, evaluated using the real queries, as the final metric.

- **Zero-shot effectiveness**: This evaluation was performed to assess the model's generalization when exposed to different distribution scenarios. Figure 11 shows the methodology used to simulate a zero-shot evaluation. Starting with all 16 selected datasets, we

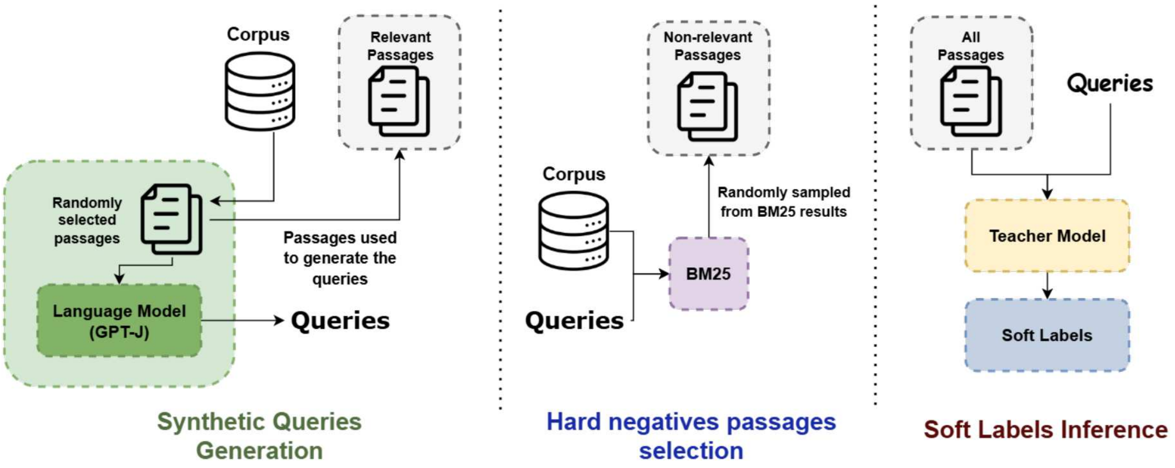
Corpus
Relevant Passages
Randomly selected passages
Passages used to generate the queries
Language Model (GPT-J)
Queries
Synthetic Queries Generation
Non-relevant Passages
Corpus
Randomly sampled from BM25 results
BM25
Queries
Hard negatives passages selection
All Passages
Queries
Teacher Model
Soft Labels
Soft Labels Inference

Figure 10: Pipeline for generating the synthetic triples ⟨query, passage, soft label⟩ for the InRanker model.

randomly selected 8 datasets and used their synthetic queries to fine-tune the model. The real queries from these 8 datasets were then used to evaluate the in-domain effectiveness of the model, while the remaining 8 datasets were used to simulate a zero-shot scenario. We repeated this process twice (sample sets 1 and 2) to reduce the possibility of the selected datasets coincidentally improving performance.

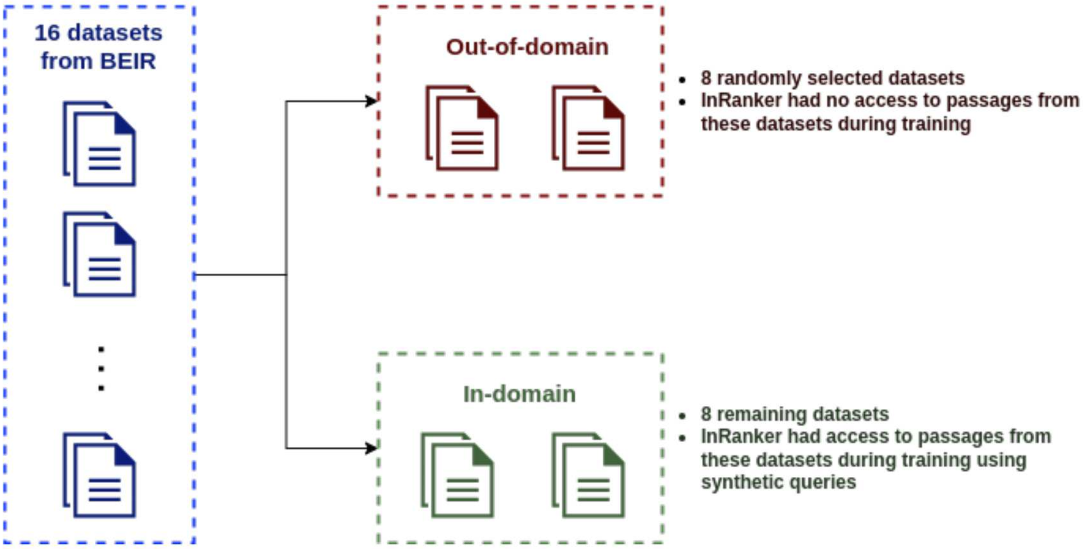
16 datasets from BEIR
Out-of-domain
8 randomly selected datasets
InRanker had no access to passages from these datasets during training
In-domain
8 remaining datasets
InRanker had access to passages from these datasets during training using synthetic queries

Figure 11: Methodology used to evaluate the effectiveness of the models in zero-shot scenarios. We created two sets, each containing 8 randomly selected datasets (in-domain) and the remaining 8 datasets to simulate an O.O.D scenario.

4 Results

4.1 Experimental Setup

Table 1 presents the hyperparameters used for training the models using an A100 GPU with 80GB of VRAM. All experiments were conducted using a learning rate of $7e^{-}5$ and AdamW as optimizer with $\beta_1 = 0.9$ e $\beta_2 = 0.99$. The batch size was set to 32. For the 3B model, we used gradient checkpointing and gradient accumulation (to achieve an effective batch size of 2×16) due to memory constraints. During the generation of soft labels using the teacher model, we sampled 9 non-relevant passages for each relevant passage, leading to 10 pairs of logits per query as shown in Figure 10. It is important to note that, differently from InPars and Promptagator, which train a separate model for each dataset, InRanker is a single model trained on all 16 datasets from BEIR simultaneously.

Parameters	**Dataset**	**Steps**	**Epochs**	**Training Duration**
60M	Human Soft	400k	10	7h
	Synthetic Soft	329k	1	5:30h
220M	Human Soft	400k	10	15h
	Synthetic Soft	329k	1	12h
3B	Human Soft	400k	10	300h
	Synthetic Soft	329k	1	250h

Table 1: Training hyperparameters and duration using an A100-80GB GPU.

During evaluation we used a strategy to simulate out-of-domain scenarios. For this, we created two samples of datasets containing synthetic queries from 8 datasets of BEIR. These sets were used during the fine-tuning process, while the remaining datasets were used to simulate in-domain scenarios as shown in Figure 11. Table 2 shows the datasets that were *randomly* chosen for inclusion in each sample set, resulting in the use of 12 out of the 16 BEIR datasets (as some were not used for training at all). We also tested different loss functions, including the KL divergence and MSE, to match the logits of the two models. Table 3 shows the results, indicating that KL divergence was slightly worse for T5-small and that using only the true label in MSE as opposed to using both true and false labels also reduced the effectiveness.

4.2 Experimental Results

We distilled monoT5-3B to models with parameters ranging from 60M to 3B, using combinations of the following configurations:

Human Hard: representing the common approach for training rankers with human-annotated hard (i.e., binary) labels from the MS MARCO passage ranking dataset. In this case, a vanilla cross-entropy loss is used:

$$\mathcal{L}_{\text{CE}} = -\log P_{\text{relevant}} - \log P_{\text{non-relevant}} \tag{12}$$

where P_{relevant} and $P_{\text{non-relevant}}$ are the probabilities assigned by the model to the relevant and non-relevant query-document pair, respectively. Non-relevant pairs are sampled from the top-1000 retrieved by BM25.

Human Soft: representing a distillation step for matching the logits of a teacher and a student model, using real (human-generated) queries from the ranking dataset as inputs, but without the binary relevance judgments for targets.

Synthetic Soft: representing a distillation step for matching the logits of the two models, similar to the previous configuration, but using exclusively synthetic queries generated from the corresponding BEIR corpora with InPars [3, 20].

From Table 4, we see that both distillation steps were essential for improving the average nDCG@10 score compared to the model trained solely using human hard labels from MS MARCO. As a result, InRanker-60M (row 3) and InRanker-220M (row 6), despite being 50x and 13x smaller than the teacher model, were able to improve their effectiveness on the BEIR benchmark significantly. Moreover, models trained exclusively on MS MARCO soft labels (rows 2 & 5) saw an increase in effectiveness in comparison to training on solely hard labels (rows 1 & 4), corroborating findings from previous studies regarding the effectiveness of soft labels [18, 19, 15, 17]. Furthermore, we observed an increase in the effectiveness even in self-distillation training (row 8), where the student learns soft labels generated by itself. We hypothesize that the improvement stems from the extra knowledge provided by the language model used to generate the synthetic queries. We did not provide results for the 3B model trained on both human soft and synthetic soft due to computational costs.

Furthermore, in Table 5, we present a effectiveness comparison between InRanker, Promptagator [14], and RankT5 [39]. Although we used monoT5-3B as a teacher for our experiments, which has a lower effectiveness on average when compared to Promptagator or RankT5-3B, our method is model-agnostic and thus one could use a stronger teacher model and anticipate even stronger results. Nonetheless, InRanker remains competitive in both model groups of 220M and 3B parameters, outperforming the other two baselines in 6 out of the 10 evaluated datasets, despite the average score not reflecting this due to

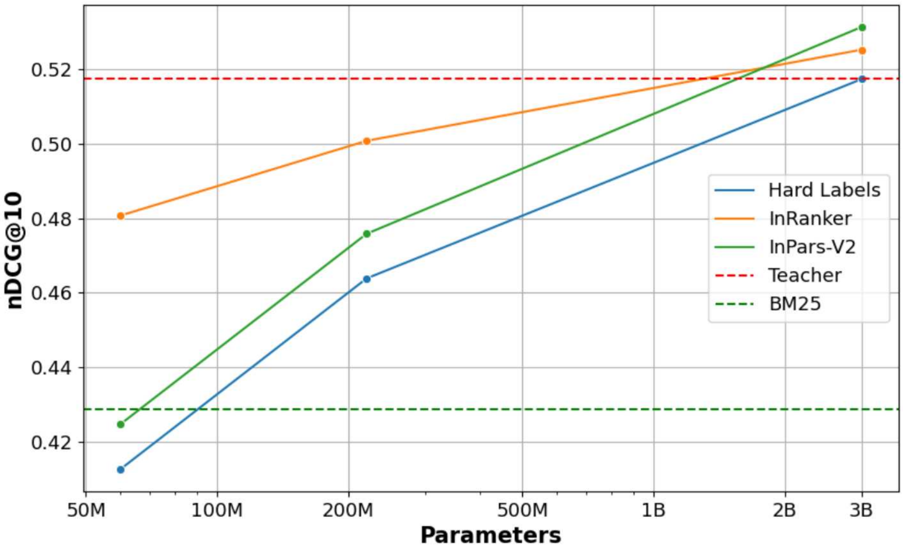

Figure 12: Effectiveness on the BEIR benchmark [36]. All models are based on monoT5 [29], applying different fine-tuning methods.

Promptagator and RankT5 attaining a significantly higher score in two datasets: ArguAna and Touché. Finally, Figure 12 presents the results of the comparison between Inranker, monoT5, Inpars-V2, and BM25 in terms of their effectiveness.

4.3 Portuguese Results

To further assess the efficacy of the technique in different languages, we evaluated InRanker on a Portuguese dataset for information retrieval: QUATI [7]. Instead of using the same T5 model, we started from PTT5 [8, 31], a Portuguese fine-tuned version of T5. We used the same two-step training approach as before, but with a strategy that allowed us to distill a Portuguese model using an English teacher (monoT5-3B). Given the availability of a translated version of MS MARCO in Portuguese [4], the first step (human soft) involved training the model using the Portuguese text, while matching the soft labels generated by the teacher using the original English text. This approach enabled us to leverage a stronger model that is not available in Portuguese for the distillation process. In the second step, involving synthetic soft labels from BEIR, we trained using the English text, as there is no translated version of BEIR available in Portuguese.

The results of this evaluation are presented in Table 6. We conclude that, similarly to English, the distillation process was able to improve the effectiveness of models in a zero-shot manner, as the models were trained using only real data from MS MARCO and synthetic data from BEIR. Remarkably, InRanker-740M surpassed the effectiveness of the mT5-3.7B on QUATI. Note that for the QUATI evaluation we used the same prompt presented in the paper to annotate all unjudged documents using gpt-4-turbo. Therefore all results are presented with a jugded@10 of 100%.

Table 7 presents the results obtained after distilling the models using soft labels from MS MARCO and BEIR. We can observe the impact of both proposed distillation steps, namely using soft human labels and soft synthetic labels, which bring significant effectiveness improvements over the base models. In particular, using logits from MS MARCO leads to an average of a 2-point nDCG@10 improvement for each model, while the subsequent fine-tuning phase with the synthetic BEIR queries further enhances their effectiveness by 4.5 points for T5-small and approximately 1.4 points for T5-base.

For the Portuguese evaluation, hypothesis testing results were provided to assess the statistical significance of the outcomes for each query. Using all 50 queries from the Quati dataset, which requires ranking $1,000$ passages per query, we applied a paired t-test since the queries and passages were identical across both models. For smaller models, a non-parametric test (Wilcoxon signed-rank test) was used because the Shapiro-Wilk test indicated that score differences were not normally distributed. The results show that all InRanker models outperformed the monoPTT5 models, as the null hypothesis (equal effectiveness) was rejected. All results are shown at Table 8.

4.4 Latency Improvement

In this subsection, we present the latency improvements achieved by distilling the knowledge of larger rankers into student models. Figure 13 illustrates the latency versus effectiveness (nDCG@10) of both InRanker and the non-distilled monoT5 equivalents. It was measured using a Tesla T4 GPU with 16GB of RAM, 1000 passages and 16 bits of precision, each with approximately 256 tokens. From the figure, we conclude that the InRanker model can achieve results comparable to larger models while being 3x faster.

4.5 Ablation Experiments

In this section, we present our ablation experiments aimed at validating the best configuration for distilling monoT5-3B into smaller T5-based models, as well as assessing their zero-shot capabilities. The initial experiments we conducted focused on evaluating how distillation would affect the model's effectiveness on novel dataset distributions that were not seen during training, i.e., we did not generate synthetic queries for them. To achieve this, we created two subsets, each containing 8 randomly selected datasets from 16 datasets of BEIR, which we named sample sets 1 and 2 and used only one set for training per experiment. The datasets that were used for training are designated as the "in-domain" category, while the remaining datasets, i.e. the other 8 datasets that are not part of the training set, represent the "out-of-domain" (O.O.D.) category.

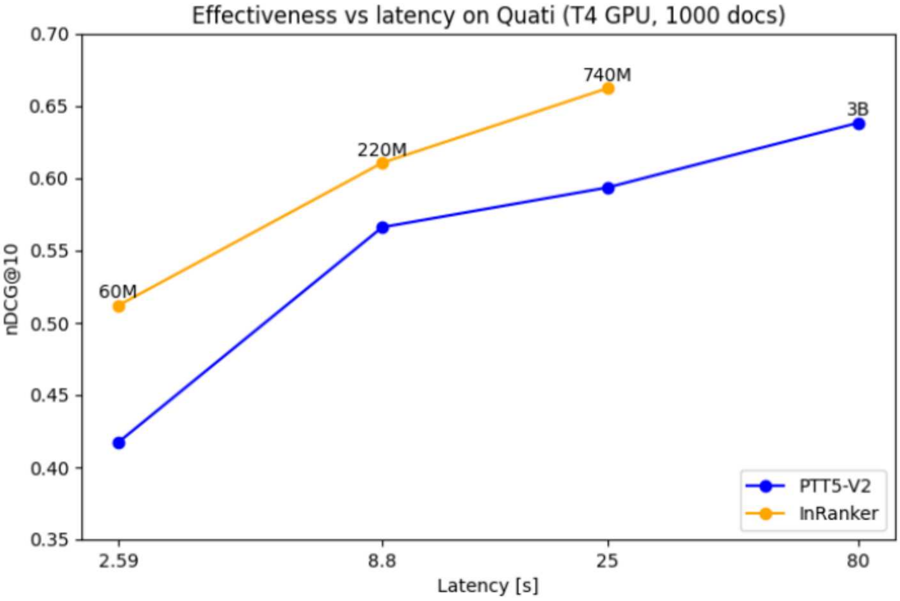

Figure 13: Latency and effectiveness comparison of InRanker and monoT5 models on Quati. Latency measured on a T4 GPU with 1000 ranked passages.

Impact of soft knowledge distillation on O.O.D. effectiveness Our first ablation experiment focused on evaluating the initial distillation process using the MS MARCO dataset with soft labels. To accomplish this, we generated logits with monoT5-3B and trained both T5-base and T5-small models for 10 epochs. As shown in Table 9, rows 1-2 & 5-6, both models demonstrated an improvement in their nDCG@10 scores compared to the baseline, which was trained using the hard labels from MS MARCO. Remarkably, the overall score increased in both scenarios, even though the models were not exposed to any BEIR passages during this phase.

Adding soft synthetic targets as a second distillation phase For the next experiment, we applied a second distillation step with synthetic soft labels on top of the model that we acquired from the last phase (monoT5 w/ soft human labels). For that, we used the 100K synthetic queries generated by InPars for each dataset indicated as “in-domain” and trained for 10 epochs. As shown in Table 9, rows 3 & 7, while it was expected that the in-domain datasets would have an increase in their nDCG@10 scores, we observe that the out-of-domain datasets also had improvements, suggesting that the model’s generalization capabilities were enhanced.

Using hard human targets for the first distillation phase Finally, we investigated the impact of skipping the first phase of distillation on MS MARCO soft labels, and instead starting from a model that was trained on hard human labels (monoT5-small and monoT5-base) and directly training using the synthetic soft BEIR targets. As we can see in Table 9, rows 3-4 & 7-8, when comparing with the model that was trained using the soft human targets, the overall effectiveness was reduced. From this, we conclude that the distillation step that includes the soft human targets on MS MARCO is beneficial, as it improves the model's effectiveness in both in-domain and out-of-domain scenarios.

Upper bound for soft distillation To estimate the upper bound of the effectiveness that these models could attain through distillation, we repeated the process using *real queries* from BEIR, (i.e., the validation queries) instead of the synthetic ones. Results presented in Table 10 show that for both model sizes, there was an increase in effectiveness for the in-domain datasets, as the model was exposed to the evaluation queries during training. However, we also observed an increase in effectiveness for out-of-domain datasets, indicating that the synthetic queries used for training could be improved.

Dataset	Sample Set 1	Sample Set 2
TREC-COVID		✓
NFCorpus	✓	
BioASQ		✓
NQ	✓	✓
HotpotQA	✓	✓
Climate-FEVER		
DBPedia	✓	
TREC-NEWS		
Robust04		✓
ArguAna		
Touché-2020		
Quora	✓	
SCIDOCS	✓	✓
SciFact		✓
FiQA-2018	✓	✓
Signal-1M	✓	

Table 2: Composition of the two sample sets used in the ablation experiments, using datasets from the BEIR benchmark.

Parameters	Loss	nDCG@10
60M	MSE with normalized logits	0.4807
	MSE with "true" logit only	0.4748
	KL divergence	0.4712
220M	MSE with normalized logits	0.5008
	KL divergence	0.5012

Table 3: Average nDCG@10 on 16 datasets of the BEIR benchmark with varying loss functions.

Model	**Training Configurations**			**Avg.**
	Human Hard	Human Soft	Synthetic Soft	**Score**
(1) monoT5-60M	✓			0.4125
(2) ↪ w/ soft human		✓		0.4356
(3) InRanker-60M		✓	✓	**0.4807**
(4) monoT5-220M	✓			0.4638
(5) ↪ w/ soft human		✓		0.4870
(6) InRanker-220M		✓	✓	**0.5008**
(7) monoT5-3B*	✓			0.5174
(8) InRanker-3B	✓		✓	**0.5253**

Table 4: Distillation results (nDCG@10) on 16 BEIR datasets. The model marked with * represents the teacher model. We did not train InRanker-3B on human soft labels due to computational constraints.

Dataset	**InRanker 60M**	**InRanker 220M**	**Promptagator++ 110M + 110M**	**RankT5-Enc 220M**	**InRanker 3B**	**monoT5 3B***	**RankT5-Enc 3B**
TREC-COVID	0.7775	**0.7984**	0.7620	0.7896	0.8175	0.7936	**0.8237**
NFCorpus	0.3547	0.3658	0.3700	**0.3731**	0.3825	0.3801	**0.3990**
HotpotQA	0.7563	**0.7742**	0.7360	0.7269	**0.7800**	0.7595	0.7536
Climate-FEVER	0.2729	**0.2914**	0.2030	0.2462	**0.2931**	0.2835	0.2753
DBPedia	0.4451	**0.4650**	0.4340	0.4373	**0.4762**	0.4719	0.4598
ArguAna	0.2466	0.2873	**0.6300**	0.3094	**0.4243**	0.3824	0.4069
Touché-2020	0.2883	0.2897	0.3810	**0.4449**	0.2924	0.3026	**0.4869**
SCIDOCS	0.1788	0.1911	**0.2010**	0.1760	**0.1990**	0.1978	0.1918
SciFact	0.7490	**0.7618**	0.7310	0.7493	**0.7831**	0.7773	0.7600
FiQA-2018	0.4043	0.4431	**0.4940**	0.4132	0.5027	**0.5068**	0.4932
Average	0.4474	0.4668	**0.4942**	0.4666	0.4951	0.4856	**0.5050**

Table 5: Comparison of the effectiveness for various reranking models, measured by nDCG@10 on the BEIR benchmark. The model marked with * represents the teacher model used for training InRanker. Bolded scores correspond to the best effectiveness on a specific dataset for a given model size, while underlined scores indicate the best effectiveness overall.

Model	Training Configurations			
	Human Hard	Human Soft	Synthetic Soft	**Avg. Score**
(1) PTT5-v2-60M	✓			0.4225
(2) ↪ w/ soft human		✓		0.4372
(3) InRanker-60M		✓	✓	**0.5121**
(4) PTT5-v2-220M	✓			0.5662
(5) ↪ w/ soft human		✓		0.5693
(6) InRanker-220M		✓	✓	**0.6108**
(7) PTT5-v2-740M	✓			0.5917
(8) ↪ w/ soft human		✓		0.6362
(9) InRanker-740M		✓	✓	**0.6624**
(10) monoT5-3B	✓			0.4864
(11) mT5-3.7B	✓			**0.6593**

Table 6: InRanker results on QUATI, a Portuguese evaluation dataset for information retrieval using PTT5-v2 [32]. All synthetic soft labels were generated using the BEIR datasets.

Dataset	**T5-small** (60M)			**T5-base** (220M)			**T5-3B**
	Baseline	1st Step	+ 2nd Step	Baseline	1st Step	+ 2nd Step	Teacher
TREC-COVID	0.6928	0.7247	**0.7775**	0.7775	0.7643	**0.7984**	0.7936
NFCorpus	0.3180	0.3475	**0.3547**	0.3570	0.3639	**0.3658**	0.3801
BioASQ	0.4880	0.4648	**0.5516**	0.5240	0.5281	**0.5652**	0.5652
NQ	0.4733	0.5214	**0.5469**	0.5674	0.5855	**0.5971**	0.6251
HotpotQA	0.5996	0.6842	**0.7563**	0.6950	0.7546	**0.7742**	0.7595
Climate-FEVER	0.2116	0.2488	**0.2729**	0.2451	0.2739	**0.2914**	0.2835
DBPedia	0.3437	0.3745	**0.4451**	0.4195	0.4446	**0.4650**	0.4719
TREC-NEWS	0.3848	0.4478	**0.4646**	0.4475	**0.4808**	0.4695	0.4806
Robust04	0.4222	0.4782	**0.5386**	0.5016	0.5588	**0.5774**	0.6171
ArguAna	0.1274	0.1098	**0.2466**	0.1946	0.2431	**0.2873**	0.3824
Touché-2020	0.2643	0.2557	**0.2883**	0.2773	**0.2991**	0.2897	0.3026
Quora	0.8259	0.8246	**0.8335**	0.8230	0.8418	**0.8427**	0.8347
SCIDOCS	0.1436	0.1526	**0.1788**	0.1649	0.1746	**0.1911**	0.1978
SciFact	0.6963	0.7022	**0.7490**	0.7356	0.7505	**0.7618**	0.7773
FiQA-2018	0.3377	0.3712	**0.4043**	0.4136	0.4374	**0.4431**	0.5068
Signal-1M	0.2711	0.2612	**0.2820**	0.2771	0.2910	**0.2926**	0.3004
Average	0.4125	0.4356	**0.4807**	0.4638	0.4870	**0.5008**	**0.5174**

Table 7: nDCG@10 values for each dataset after two steps of distillation.

Models	**t-value**	**p**	**w-statistic**	**p**
InRanker-small monoPTT5-v2-small	-	-	201	0.0001
InRanker-base monoPTT5-v2-base	2.607	0.0121	-	-
InRanker-large monoPTT5-v2-large	3.601	0.0007	-	-

Table 8: Hypothesis testing of means using a two-tailed paired t-test on nDCG@10 values for each query (50 queries) from the Quati dataset. For InRanker-small, due to the non-normality of the differences (a condition required for the t-test), we used a non-parametric test (Wilcoxon Signed-Rank test).

T5 Model	Training Configurations			Sample Set 1		Sample Set 2	
	Human Hard	Human Soft	Synthetic Soft	In-domain	O.O.D.	In-domain	O.O.D.
(1) 60M (monoT5)	✓			0.4141	0.4109	0.4817	0.3434
(2) 60M		✓		0.4422	0.4290	0.5124	0.3587
(3) 60M (InRanker)		✓	✓	**0.4768**	**0.4716**	**0.5558**	**0.3852**
(4) 60M	✓		✓	0.4475	0.4587	0.5355	0.3617
(5) 220M (monoT5)	✓			0.4647	0.4629	0.5475	0.3801
(6) 220M		✓		0.4867	0.4873	0.5692	0.4048
(7) 220M (InRanker)		✓	✓	**0.4945**	**0.5028**	**0.5874**	**0.4083**
(8) 220M	✓		✓	0.4905	0.4942	0.5832	0.3941
(9) 3B* (monoT5)	✓			0.5095	0.5253	0.6053	0.4295

Table 9: Comparison of the in-domain vs out-of-domain effectiveness of our method, measured by nDCG@10. The model marked with * represents the teacher model used for the knowledge distillation process.

Model	Sample Set 1		Sample Set 2	
	In-domain	O.O.D.	In-domain	O.O.D.
InRanker-60M	0.4768	0.4716	0.5558	**0.3852**
↪ w/ real queries	**0.4975**	**0.4719**	**0.5860**	0.3813
InRanker-220M	0.4945	0.5028	0.5874	0.4083
↪ w/ real queries	**0.5242**	**0.5175**	**0.6159**	**0.4202**

Table 10: Upper bound effectiveness (nDCG@10) using real queries from BEIR for the distillation datasets. Bold indicates the best between using synthetic and real queries.

5 Conclusion

This master's thesis introduces a method for distilling the knowledge of information retrieval models and improve upon previous work on how to better use synthetic data, aimed at improving the out-of-domain effectiveness of students. The methodology involves two steps of distillation: (1) using a human-curated corpus, and (2) using synthetic data generated by an LLM. The first step is used to make the model learn about the retrieval task, whereas the second step is responsible to specialize the model on target domains. Ablation studies showed that these specializations not only improved the model's effectiveness on the desired domain but also enhanced its generalization capabilities when evaluated on out-of-domain datasets. Additionally, our work shows that it is possible to improve a reranker's capabilities in specific domains without the need for additional human-annotated labels, since the queries used to specialize the model were generated synthetically. However, we observe that synthetic query generation could be improved since the real queries achieved a better out-of-domain effectiveness compared to the model trained solely on synthetic ones.

Our study reveals that, through this knowledge distillation process, smaller models can achieve results comparable to the teacher (monoT5-3B) or larger models such as monoT5-large, despite being an order of magnitude smaller. In terms of latency, moving from a model with 3B parameters to a 220M model (InRanker-base) can reduce latency by up to 10 times. Furthermore, the approach can be applied in the context of multilingual transfer knowledge, where an English teacher was used to generate soft labels to train a Portuguese fine-tuned T5 (PTT5). With this, we were able to surpass the zero-shot effectiveness of a multilingual version of monoT5-3B (mT5-3b) that was tuned to English and Portuguese with a model 5x smaller (PTT5v2-large). Therefore, the method is particularly significant for applications where computational resources are limited, in production environments, or for languages with a lack of available models that can serve as a teacher.

5.1 Limitations

In this master's thesis, the proposed methodology was applied to rerankers to improve their effectiveness in out-of-domain scenarios. However, due to the nature of the loss function, it is challenging to train a dense retriever using the same technique, as it typically relies on contrastive loss. Since dense retrievers process the query and passage at different stages, they are usually trained with a significantly larger batch size, which is increased through in-batch negative samples. Another limitation of this work is related to the evaluation datasets. Although we separated 8 datasets for fine-tuning with synthetic queries, we cannot guarantee that the remaining 8 datasets of each set, which simulate out-of-domain scenarios, do not have a similar composition.

References

[1] I. A. Ajah and H. F. Nweke. Big data and business analytics: Trends, platforms, success factors and applications. *Big Data and Cognitive Computing*, 3(2), 2019. ISSN 2504-2289. doi: 10.3390/bdcc3020032. URL https://www.mdpi.com/2504-2289/3/2/32.

[2] M. Alaofi, L. Gallagher, M. Sanderson, F. Scholer, and P. Thomas. Can generative llms create query variants for test collections? an exploratory study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 1869–1873, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591960. URL https://doi.org/10.1145/3539618.3591960.

[3] L. Bonifacio, H. Abonizio, M. Fadaee, and R. Nogueira. InPars: Unsupervised Dataset Generation for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 2387–2392, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531863. URL https://doi.org/10.1145/3477495.3531863.

[4] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira. mmarco: A multilingual version of the ms marco passage ranking dataset, 2022. URL https://arxiv.org/abs/2108.13897.

[5] L. Boytsov, P. Patel, V. Sourabh, R. Nisar, S. Kundu, R. Ramanathan, and E. Nyberg. Inpars-light: Cost-effective unsupervised training of efficient rankers, 2023.

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[7] M. Bueno, E. S. de Oliveira, R. Nogueira, R. A. Lotufo, and J. A. Pereira. Quati: A brazilian portuguese information retrieval dataset from native speakers, 2024. URL https://arxiv.org/abs/2404.06976.

[8] D. Carmo, M. Piau, I. Campiotti, R. Nogueira, and R. Lotufo. Ptt5: Pretraining and validating the t5 model on brazilian portuguese data, 2020. URL https://arxiv.org/abs/2008.09144.

[9] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du,

B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways, 2022.

[10] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models, 2022.

[11] N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the trec 2020 deep learning track, 2021.

[12] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and J. Lin. Overview of the TREC 2021 deep learning track. In I. Soboroff and A. Ellis, editors, *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021, online, November 15-19, 2021*, volume 500-335 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2021. URL `https://trec.nist.gov/pubs/trec30/papers/Overview-DL.pdf`.

[13] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, J. Lin, E. M. Voorhees, and I. Soboroff. Overview of the TREC 2022 deep learning track. In I. Soboroff and A. Ellis, editors, *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, online, November 15-19, 2022*, volume 500-338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022. URL `https://trec.nist.gov/pubs/trec31/papers/Overview_deep.pdf`.

[14] Z. Dai, V. Y. Zhao, J. Ma, Y. Luan, J. Ni, J. Lu, A. Bakalov, K. Guu, K. Hall, and M.-W. Chang. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=gmL46YMpu2J`.

[15] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2353–2359, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531857. URL `https://doi.org/10.1145/3477495.3531857`.

[16] Y. Fu, H. Peng, L. Ou, A. Sabharwal, and T. Khot. Specializing smaller language models towards multi-step reasoning. In A. Krause, E. Brunskill, K. Cho,

B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/fu23d.html`.

[17] H. Hashemi, Y. Zhuang, S. S. R. Kothur, S. Prasad, E. Meij, and W. B. Croft. Dense retrieval adaptation using target domain description. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '23, page 95–104, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700736. doi: 10.1145/3578337.3605127. URL `https://doi.org/10.1145/3578337.3605127`.

[18] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[19] S. Hofstätter, S. Althammer, M. Schröder, M. Sertkan, and A. Hanbury. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation, 2020.

[20] V. Jeronymo, L. Bonifacio, H. Abonizio, M. Fadaee, R. Lotufo, J. Zavrel, and R. Nogueira. Inpars-v2: Large language models as efficient dataset generators for information retrieval, 2023.

[21] O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020. URL `https://arxiv.org/abs/2004.12832`.

[22] T. Laitz, K. Papakostas, R. Lotufo, and R. Nogueira. Inranker: Distilled rankers for zero-shot information retrieval, 2024. URL `https://arxiv.org/abs/2401.06910`.

[23] J. Lin, R. Nogueira, and A. Yates. Pretrained transformers for text ranking: Bert and beyond, 2021. URL `https://arxiv.org/abs/2010.06467`.

[24] L. C. Magister, J. Mallinson, J. Adamek, E. Malmi, and A. Severyn. Teaching small language models to reason. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.151. URL `https://aclanthology.org/2023.acl-short.151`.

[25] A. Muhamed, I. Keivanloo, S. Perera, J. Mracek, Y. Xu, Q. Cui, S. Rajagopalan, B. Zeng, and T. Chilimbi. Ctr-bert: Cost-effective knowledge distillation for billion-parameter teacher models. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*, 2021.

[26] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, J. Heidecke, P. Shyam, B. Power, T. E. Nekoul, G. Sastry, G. Krueger, D. Schnurr, F. P. Such, K. Hsu, M. Thompson, T. Khan, T. Sherbakov, J. Jang, P. Welinder, and L. Weng. Text and code embeddings by contrastive pre-training, 2022.

[27] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, 2016.

[28] J. Ni, C. Qu, J. Lu, Z. Dai, G. Hernandez Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, and Y. Yang. Large dual encoders are generalizable retrievers. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.669. URL `https://aclanthology.org/2022.emnlp-main.669`.

[29] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.63. URL `https://aclanthology.org/2020.findings-emnlp.63`.

[30] G. Penha, E. Palumbo, M. Aziz, A. Wang, and H. Bouchard. Improving content retrievability in search with controllable query generation. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3182–3192, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583261. URL `https://doi.org/10.1145/3543507.3583261`.

[31] M. Piau, R. Lotufo, and R. Nogueira. ptt5-v2: A closer look at continued pretraining of t5 models for the portuguese language, 2024. URL `https://arxiv.org/abs/2406.10806`.

[32] M. Piau, R. Lotufo, and R. Nogueira. ptt5-v2: A closer look at continued pretraining of t5 models for the portuguese language, 2024. URL `https://arxiv.org/abs/2406.10806`.

[33] R. Pradeep, R. Nogueira, and J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models, 2021.

[34] R. Ren, Y. Qu, J. Liu, X. Zhao, Q. Wu, Y. Ding, H. Wu, H. Wang, and J.-R. Wen. A thorough examination on zero-shot dense retrieval. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15783–15796, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1057. URL `https://aclanthology.org/2023.findings-emnlp.1057`.

[35] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira. No parameter left behind: How distillation and model size affect zero-shot retrieval, 2022.

[36] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and*

Benchmarks Track (Round 2), 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

[37] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[38] W. Yang, K. Lu, P. Yang, and J. Lin. Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19. ACM, July 2019. doi: 10.1145/3331184.3331340. URL http://dx.doi.org/10.1145/3331184.3331340.

[39] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, and M. Bendersky. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2308–2313, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3592047. URL https://doi.org/10.1145/3539618.3592047.