Exploring Neural IR Approaches in Europeana

Unlocking Multilingual Insights for Cultural Heritage Search

Suhaib Basir

Delft University of Technology

TU Delft

Exploring Neural IR Approaches in Europeana

Unlocking Multilingual Insights for Cultural Heritage Search

by

Suhaib Basir

Suhaib
Basir

Instructor:	J. Urbano
Instructor:	M. Marrero
Project Duration:	March, 2024 - February, 2025
Faculty:	Faculty of EEMCS, Delft

Cover:	Vishnu Mohanan
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

I would like to express my deepest gratitude to everyone who has supported me throughout this journey.

First and foremost, to my thesis supervisor, Julian Urbano, and my company supervisor at Europeana, Monica Marrero, without whom none of this would have been possible. Their expertise, guidance, and feedback have been invaluable throughout every stage of my thesis, from the initial conceptualization to the final execution. They were always available whenever I needed their help, providing advice and constructive criticism that helped me refine my ideas and overcome many obstacles. Beyond their academic mentorship, their encouragement and belief in my abilities gave me the confidence to push forward, even in the face of many challenges. For their generosity, knowledge, wisdom, and patience, I will always be grateful.

I would also like to thank all of my friends and classmates I have had the privilege of working with over the past two and a half years at TU Delft. I also want to express my gratitude to the RND team at Europeana for their support when I first joined the company and for their bi-weekly encouragement during our stand-up meetings.

Finally, I would like to thank my parents and my sister for always supporting me no matter what and having the faith in me to do well.

This thesis represents nearly a year of dedication and learning. The experience, skills, and insights gained throughout this journey have been invaluable, shaping my personal and professional growth. I will carry these lessons with me for the rest of my life.

With this, I present my thesis on Exploring Neural Information Retrieval in Europeana, a collaborative project between TU Delft and Europeana. This thesis offers a comprehensive investigation into the application of Neural Information Retrieval techniques within Europeana's digital library. This research contributes by developing a preprocessed dataset for this task and provides a structured evaluation of multilingual Neural Information retrieval models for Europeana's vast data collection, laying the groundwork for future advancements in Europeana's search infrastructure.

Suhaib Basir
Delft, February 2025

Abstract

Europeana is a digital library of Europe's cultural heritage, housing a large corpus of data representing artworks, literature, historical locations and many culturally significant items. Europeana currently relies of traditional text-matching retrieval, such as BM25, to facilitate their search and discovery across millions of multilingual metadata-based records. However, these models are not capable of semantic understanding and require additional treatments to facilitate multilingual retrieval which costs Europeana resources, these treatments entail translating queries and data from other language into English and enriching content by adding entities from linked open data. Europeana's current methodology is ultimately limited in its ability to provide semantically relevant multilingual search results.

This thesis investigates the application of Neural Information Retrieval (NIR) to enhance Europeana's search capabilities. This investigation aims to assess the impact of NIR on multilingual retrieval and retrieval performance while also determining the value of existing translation and enrichment processes. To support this investigation, we contribute by developing a structured and preprocessed dataset specifically for NIR, as no such dataset previously existed for NIR. We conduct an extensive evaluation of NIR models, analyzing the impact of fine-tuning, query treatments, and document treatments on retrieval quality. Additionally, we assess the computational requirements, scalability, and practicality of deploying NIR, identifying trade-offs in retrieval efficiency and resource consumption, to provide an idea of an infrastructure Europeana would need to implement NIR.

This research required meticulous planning across all stages—from data collection and formatting to model training and evaluation—since applying Neural IR at this scale for metadata search is new for Europeana. Therefore, research not only provides insights into the viability of NIR as a replacement or enhancement to Europeana's existing search system but also lays the foundation for future advancements in multilingual retrieval for Europeana.

Through this thesis, we found that Neural IR models can offer promising improvements in multilingual retrieval and semantic search, reducing reliance on exact term matching. Our analysis suggests that not all of Europeana's current preprocessing treatments are necessary for NIR models, as they inherently capture cross-lingual relationships more effectively than BM25, though the benefits vary depending on the model and configuration used. Overall, we recommend that a hybrid retrieval system that leverages both lexical and neural approaches may be the most practical solution for Europeana and warrants further exploration.

The integration of NIR presents several challenges, particularly in terms of infrastructure and evaluation. NIR models are sensitive to changes in document structure and content, requiring careful consideration of indexing and fine-training. Furthermore, while these models improve semantic search, they may struggle with entity-based queries, where BM25's exact matching approach remains valuable.

A major limitation of this study was the absence of explicit relevance judgements in our dataset, which constrained our ability to make definitive conclusions about retrieval effectiveness. Future work should prioritize the development of a comprehensive evaluation framework, incorporating expert and user-based relevance assessments, to enable a more robust analysis of NIR's impact.

Contents

1

Introduction

1.1. Multilingual search in Europeana

Europeana is a digital library aggregating cultural heritage content from libraries, archives, and museums across Europe, providing access to a vast and diverse collection. This collection contains various media types, languages, and metadata quality, often enriched with additional entity-based information from supplemental sources like Wikipedia, and in some cases data is also translated into English; coming from the Europeana Translate project [1]. Europeana stands as a platform for showcasing Europe's shared cultural heritage. By digitizing and aggregating content from diverse institutions, Europeana empowers researchers, educators, and individuals to learn about and explore Europe's rich history and culture.

Currently, Europeana is powered by the Solr engine with the BM25 algorithm as its current search method. While being able to handle large-scale indexing and retrieval efficiently, the Solr engine with BM25 has limitations when addressing multilingual data. BM25, primarily a keyword-based search algorithm, struggles to account for semantic relationships, contextual understanding, and multilingual retrieval [35], which are critical for a collection as diverse as Europeana's. This creates challenges in delivering highly relevant and nuanced search results, especially when queries involve different languages or lacks explicit keyword matches. Exploring more advanced methods, such as Neural Information Retrieval (NIR) models, could potentially overcome these limitations and provide a more accurate search experience for diverse users.

Europeana's metadata is presented in more than 40 languages, from around Europe. The collection's heterogeneity extends to varying metadata quality types [2]. Many items are classified into content tiers (0 to 4) and metadata tiers (A to C) based on the level of detail in the data. These tiers help standardize and differentiate the depth of information.

Metadata files contain details such as media type, dimensions, color components, orientation, language labels, creation dates, and links to related resources. Items are also enriched with entities such as names, dates, and geographical locations. This rich metadata enables more meaningful and context-aware retrieval within Europeana's collections. All data is standardized through the Europeana Data Model (EDM) [3]. Figure 1.1 illustrates an example of the raw Europeana metadata.

Europeana conducts two types of augmentations on their data: They enrich some of the metadata fields from sources such as Wikipedia; these are entity based, such as adding names, places, locations, etc identified in the object in other languages. Additionally some other fields are even translated into English, if it is another language. These methods are quite costly to Europeana as they have over 62 million documents and the need to enrich and translate all of them would cost time and considerable resources.

[1] https://pro.europeana.eu/project/europeana-translate
[2] https://pro.europeana.eu/post/publishing-framework
[3] https://pro.europeana.eu/page/edm-documentation

```
<ore:Proxy rdf:about="http://data.europeana.eu/proxy/europeana/2021672/
    resource_document_mauritshuis_670">
<dc:date rdf:resource="#1665%2F1665"/>
<dc:identifier>resource_document_mauritshuis_670</dc:identifier>
<dcterms:medium rdf:resource="http://data.europeana.eu/concept/2728"/>
<edm:europeanaProxy>true</edm:europeanaProxy>
<ore:proxyFor rdf:resource="http://data.europeana.eu/item/2021672/
    resource_document_mauritshuis_670"/>
<ore:proxyIn rdf:resource="http://data.europeana.eu/aggregation/europeana/2021672/
    resource_document_mauritshuis_670"/>
<ore:lineage rdf:resource="http://data.europeana.eu/proxy/provider/2021672/
    resource_document_mauritshuis_670"/>
<edm:type>IMAGE</edm:type>
</ore:Proxy>
<ore:Proxy rdf:about="http://data.europeana.eu/proxy/provider/2021672/
    resource_document_mauritshuis_670">
<dc:creator>Johannes Vermeer</dc:creator>
<dc:date>1665 - 1665</dc:date>
<dc:description xml:lang="nl-NL">Meisje met de parel is het beroemdste schilderij van Vermeer
    . Het is geen portret, maar een    tronie   : een fantasiekop. Tronies beelden een
    bepaald type of karakter uit, in dit geval een meisje in exotische kledij, met een
    oosterse tulband en een onwaarschijnlijk grote parel in het oor. Vermeer was de meester
    van het licht. Hier is dat te zien aan het zachte in het meisjesgezicht, de glimlichtjes
    op haar vochtige lippen. En aan de glanzende parel.</dc:description>
<dc:description xml:lang="en-GB">Girl with a Pearl Earring is Vermeers most famous
    painting. It is not a portrait, but a    tronie        a painting of an imaginary figure.
     Tronies depict a certain type or character; in this case a girl in exotic dress, wearing
     an oriental turban and an improbably large pearl in her ear. Johannes Vermeer was the
    master of light. This is shown here in the softness of the girls face and the glimmers
     of light on her moist lips. And of course, the shining pearl.</dc:description>
<dc:format>65 cm</dc:format>
<dc:format>39 cm</dc:format>
<dc:format>44.5 cm</dc:format>
<dc:format>74 cm</dc:format>
<dc:identifier>670</dc:identifier>
<dc:title xml:lang="nl-NL">Meisje met de parel</dc:title>
<dc:title xml:lang="en-GB">Girl with a Pearl Earring</dc:title>
<dc:type xml:lang="nl-NL">schilderij</dc:type>
<dc:type xml:lang="en-GB">painting</dc:type>
<dcterms:medium xml:lang="nl-NL">doek</dcterms:medium>
<dcterms:medium xml:lang="en-GB">canvas</dcterms:medium>
<edm:europeanaProxy>false</edm:europeanaProxy>
<ore:proxyFor rdf:resource="http://data.europeana.eu/item/2021672/
    resource_document_mauritshuis_670"/>
<ore:proxyIn rdf:resource="http://data.europeana.eu/aggregation/provider/2021672/
    resource_document_mauritshuis_670"/>
<edm:type>IMAGE</edm:type>
</ore:Proxy>
```

Figure 1.1: Example of Europeana XML metadata of "Girl with a Pearl Earring" by Vermeer

1.2. Background

1.2.1. Information retrieval

Information Retrieval (IR) is the task of finding relevant information or documents from an extensive collection based on a user's query [38]. The primary objective of IR systems is to rank documents in order of their relevance to the query. IR has been foundational to search engines, recommendation systems, and knowledge discovery platforms, among other applications. Traditional IR approaches use lexical matching, statistical methods, and heuristic-based algorithms to connect queries with relevant documents. These systems typically compare the terms in a query with those in the document collection to measure relevance. However, such methods are inherently limited in understanding the semantics of queries and documents, mainly when dealing with natural language queries and multilingual data.

1.2.2. Mono-, Cross-, Multilingual-Information retrieval

Monolingual Information Retrieval (IR) refers to the retrieval of information where both the query and the documents are in the same language [28]. It is the traditional form of IR, where models are trained and optimized to handle and rank documents written in a single language.

Cross-lingual Information Retrieval (CLIR) involves retrieving documents written in a different language than the query. This approach is crucial in scenarios where users may need to access information in multiple languages but can only query in one language. CLIR systems often rely on translation mechanisms, such as neural machine translation, to bridge the language gap between queries and documents, enabling users to retrieve relevant information from a multilingual corpus [28].

Multilingual Information Retrieval (MLIR) extends this concept further by enabling the retrieval of documents across multiple languages, regardless of the query language. MLIR systems are designed to simultaneously understand and process queries and documents in various languages, often leveraging multilingual large language models like multilingual-BERT or XLM-Roberta. These systems can handle multiple languages within the same retrieval process, providing a more inclusive and comprehensive search experience for users [28].

1.3. Document search at Europeana

Europeana's current search functionality is primarily monolingual, relying on the BM25 algorithm for keyword-based retrieval. To address the multilingual nature of its collection, Europeana has implemented a pilot project on its Spanish portal only available to registered users [4], utilizing English as a pivot language. It is important to note that the enrichment process is always done, but the translation has been done only on a part of the collection (in the framework of a specific project). In this approach, user queries in other languages are translated into English to retrieve documents with English metadata, aiming to surface relevant results that might not be available in the original language of the query. However, this method introduces challenges:

- Noisy Query Translations: Translating queries without sufficient context can lead to inaccuracies, resulting in less relevant search results.
- Costly and Incomplete Document Translations: Translating the extensive and ever-growing collection is resource-intensive, and incomplete translations can hinder the retrieval of important information.
- Contextual Limitations: When using a lexical matching algorithm such as BM25, context is not captured, and this is amplified by translating words into English, leading to potential misinterpretations.

These issues were highlighted in a study [22] by Europeana which evaluated the effectiveness of the pivot language approach and identified areas for improvement.

The core research challenge is to explore how Europeana's search functionality can be enhanced to more effectively handle its multilingual collection. This investigation will focus on introducing end-to-end multilingual NIR models to the Europeana dataset and evaluate their efficiency and effectiveness compared to Europeana's current approach based on BM25. This study will assess the impact of the metadata enrichment's and english-translations on retrieval quality while determining whether if NIR can improve search and reduce Europeana's reliance on these costly methods. Additionally, the research will examine the trade-offs between these models' efficiency and effectiveness while providing insights an infrastructure that is well suited for this task. It is important to note that although there is full-text included in the Europeana collection for some types of items (e.g. newspapers), the data we will be dealing with contains only metadata because the search on this type of data is the main service Europeana provides.

By leveraging the semantic understanding capabilities of NIR, the thesis aims to investigate a more effective solution for managing and searching Europeana's collection.

[4] https://www.europeana.eu/es

1.4. Research Scope

1.4.1. Objectives

The primary objective of this project is to explore how NIR approaches can be used for the search functionality of Europeana's digital library. This involves investigating various NIR models and techniques, experimenting with various approaches, assessing their applicability within Europeana's needs, evaluating their performance against the current BM25-based method, and examining whether the enrichment's and translations can be made redundant using NIR.

Furthermore, this investigation serves as Europeana's initial step into NIR. Since their current data structure is not optimized for NIR, a key aspect of this research will be developing a methodology for transforming Europeana's XML documents (as seen in figure 1.1) into a format suitable for indexing, training, and querying with NIR models. This will involve structuring and curating the dataset to align with the requirements of modern retrieval techniques while preserving the integrity and richness of Europeana's metadata.

Therefore, in this study we will:

- Investigate various state-of-the-art NIR models and the application of these models into Europeana's extensive collection.
- Examine the applicability of these models and analyse the results in terms of efficiency and identify the requirements needed to scale the specific approaches tested to the Europeana use case
- Evaluate the performance of NIR for Europeana's collection, as the focus will be on assessing its effectiveness in terms of multilinguality, relevance, and the impact of Europeana's query and data augmentation. This analysis will help determine how well NIR models handle Europeana's diverse, multilingual content and whether enrichments and translations are necessary to improve search with NIR.
- Building a dataset for NIR: Europeana's existing data is structured for Solr, which is not directly compatible with NIR models. To enable indexing, training, and querying with NIR, a key objective of this investigation is to develop a methodology for processing and reformatting the dataset for N. This includes extracting relevant fields, structuring passages for embedding-based retrieval, aligning queries with corresponding documents, and ensuring multilingual consistency.

1.4.2. Research questions

To guide this exploration, the following research questions(s) will be addressed:

How can NIR improve Europeana's search by handling multilingual and diverse metadata more effectively than the current BM25-based approach?

The following subquestions will be addressed throughout the investigation to thoroughly answer the main research question.

Subquestions:

- Subquestion 1 pertains to quality of search: How do different NIR models and treatments—particularly the use of translation, enrichment stages, and fine-tuning on Europeana's dataset—impact retrieval performance compared to the BM25-based approach?
- Subquestion 2 pertains to the implementation of NIR within Europeana: What are the infrastructural and efficiency considerations for implementing NIR in Europeana?

1.4.3. Contributions

Overall, our contributions include:

- Development of structured datasets for NIR in Europeana, along with a methodology for extracting and preparing these datasets for indexing, training, and querying.
- Evaluation of NIR models concerning model selection, fine-tuning strategies, query augmentations, and document augmentations, assessing their impact on retrieval quality.

- Examination of the effectiveness and efficiency of deploying NIR models, including an analysis of computational requirements and infrastructure needs for implementation in Europeana.

2 Literature

2.1. Search in Information retrieval

Search in information retrieval involves retrieving relevant documents from a collection based on user queries. The core principle underlying IR is the representation of queries and documents in a form that enables an efficient similarity computation. The choice of representation and similarity function varies between classical and neural approaches, leading to distinct retrieval mechanisms and performance characteristics.

2.1.1. Classical

Data representation in classical IR

Classic IR systems rely on lexical representations, where documents and queries are represented based on their explicit word occurrences. These representations do not consider word meanings and focus on the frequency and distribution of terms within a document.

One of the most fundamental lexical representations is the Bag-of-Words (BoW) model [38], which treats each document as an unordered set of words, disregarding syntax and word order. This simplistic representation enables computational efficiency but loses contextual and semantic information.

To refine lexical representations, weighting schemes like Term Frequency-Inverse Document Frequency (TF-IDF) [33] are applied. TF-IDF assigns importance to terms based on two factors:

- **Term Frequency (TF)**: Measures how often a term appears in a document.
- **Inverse Document Frequency (IDF)**: Downweights common terms by assigning higher importance to rare but meaningful words.

TF-IDF enhances retrieval by prioritizing distinguishing terms over common ones, making it a cornerstone of traditional IR ranking techniques.

To facilitate efficient search, classical IR relies on inverted indexes, a data structure that maps terms to the documents in which they appear. This indexing strategy significantly accelerates retrieval by allowing systems to directly access relevant documents without scanning the entire corpus. However, due to their reliance on exact term matching, inverted indexes struggle with challenges such as synonymy (different words with the same meaning) and polysemy (words with multiple meanings), leading to retrieval limitations.

Another representation of queries and documents in classic IR is the Vector Space Model (VSM) [24]. In such systems, both queries and documents are represented as TF-IDF weighted vectors, and their relevance is determined by measuring the similarity of the vectors. This ensures that documents with similar term distributions rank higher, regardless of length differences.

Search and similarity in Classical IR
Lexical matching forms the backbone of classical IR systems. It involves matching query terms to document terms based on their surface forms, making it straightforward and computationally efficient.

Cosine similarity [37] is a fundamental in classical IR, particularly in Vector Space Models (VSM), where both queries and documents are represented as high-dimensional vectors. It calculates the cosine of the angle between these vectors, determining how similar they are based on term overlap and distribution.

One of the most widely used lexical matching algorithms is BM25 (Best Match 25) [34], a probabilistic ranking function that scores documents by considering TF-IDF, and document length normalization.

BM25 calculates the relevance of a document D to a query Q using the following formula:

$$\text{Score}(D, Q) = \sum_{t \in Q} \text{IDF}(t) \cdot \frac{f(t, D) \cdot (k_1 + 1)}{f(t, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

where $f(t, D)$ is the term frequency of t in D, $|D|$ is the document length, avgdl is the average document length in the collection, and k_1 and b are tunable parameters. The Inverse Document Frequency (IDF) is used to penalize terms that appear in many documents, reducing their influence.

2.1.2. Neural
While classical IR techniques like TF-IDF and BM25 provide strong baselines, they rely on exact term matching and cannot capture contextual or semantic relationships between words. As a result, they struggle with synonyms, polysemy, and multilingual search. NIR addresses these challenges by leveraging deep learning models to create representations that capture semantic meaning beyond surface-level term matching.

Data representation in NIR
An embedding in natural language processing (NLP) is a technique that represents words, phrases, or entire texts as vectors of real numbers in a continuous vector space [6]. This representation captures deeper semantic information allowing words with similar meanings or functions to be positioned close to each other in this space.

Embeddings are fundamental in NLP and IR as they enable machine learning models to process and understand text data more effectively by converting text into a numerical form that models can interpret.

Embedding generation approaches have significantly advanced with integrating deep learning models, particularly those based on transformers like BERT (Bidirectional Encoder Representations from Transformers) [12]. These models operate on tokens, which are the fundamental units of text representation. A token can be a word, subword, or even a character, depending on the tokenization method used. Tokenization breaks text into these smaller units before converting them into vector embeddings. BERT has revolutionized IR by learning contextual relationships between words through bidirectional processing, capturing the full context of a word by considering the words that come before and after it [12]. DistilBERT is a streamlined version of BERT, designed to be smaller, faster, and more efficient while retaining much of BERT's capabilities [36].

Vector storage systems such as Milvus and Faiss are designed to store embeddings in a manner that allows them to be retrieved at scale.

Milvus is an open-source vector database that supports various index types and metrics, making it ideal for handling large-scale data with real-time query performance. Milvus can manage high-dimensional vectors (up to 32,000+ dimensions), which benefits Europeana's extensive collection.

FAISS, developed by Meta AI, excels at high-dimensional vector similarity searches, especially with GPU acceleration. It is most suitable for tasks where efficiency is paramount, such as multi-vector searches. While FAISS has no fixed upper limit on vector dimensions, its performance and scalability is primarily determined by the available memory and computational resources of the machine(s) it is running on.

To facilitate quick searches, indexes are created that organize vectors in a way that allows the system to rapidly find and compare them. Techniques like Approximate nearest neighbor (ANNs) algorithms

are used to enhance efficiency by reducing search complexity and enabling fast similarity comparisons in high-dimensional spaces [30]. Instead of exhaustively comparing a query vector with all stored embeddings, ANN methods approximate the nearest neighbors by first partitioning the search space and then searching only the most relevant regions.[30].

Search and similarity in NIR
Vector search involves comparing vectors to determine their similarity to one another. Given a textual query, the query is first embedded using the same embedding model as the indexed data. The search process then finds vectors in the database that are most similar to the query vector based on a similarity measure such as cosine similarity, inner product, or Euclidean distance [41].

As mentioned earlier, cosine similarity measures the angle between two vectors, making it useful for text-based retrieval where magnitude differences are less relevant. A higher cosine similarity score indicates that two vectors point in the same direction, meaning they are semantically similar. Inner product similarity, on the other hand, directly measures the dot product between vectors, emphasizing magnitude and direction. This makes it useful when vector norms carry meaning, such as ranking document relevance in neural retrieval models. [41].

ANN algorithms play a crucial role in optimizing this process. Instead of comparing the query vector to every vector in the dataset, ANN techniques group similar vectors into clusters using similarity metrics. At search time, the system narrows down the search space to a cluster of vectors likely to be relevant to the query, significantly speeding up the retrieval process while maintaining acceptable accuracy [30].

A crucial ranking function in NIR is MaxSim, a method often used in neural retrieval. Unlike traditional retrieval methods that compare entire document embeddings, MaxSim operates at the token-level, identifying the most relevant passage by maximizing the similarity between individual query and document token embeddings. This fine-grained approach enhances retrieval precision, especially in cases where different parts of a document contribute differently to relevance.

The combination of efficient vector storage, optimized search techniques, and fine-grained ranking methods enables modern NIR systems to handle large-scale, high-dimensional data efficiently.

Sparse and dense embeddings
Sparse embeddings represent text using high-dimensional vectors with many zero entries, traditionally generated through TF-IDF methods [38]. These embeddings are computationally efficient and interpretable, making them well-suited for traditional IR tasks. However, sparse embeddings struggle to capture the semantic relationships between terms, as they rely solely on the frequency of word co-occurrence within a document or corpus [27].

Dense embeddings, on the other hand, leverage neural network models to encode text into continuous vector representations [38]. These vectors are more detailed and capture more information than sparse vectors. Dense embeddings enable IR systems to retrieve documents based on semantic similarity rather than exact keyword matches, allowing for more nuanced and accurate retrieval [38].

The shift from sparse to dense embeddings has marked a significant paradigm shift in IR, enabling models to understand deeper relationships within text. NIR methods build on this foundation by combining dense embeddings with advanced deep learning architectures to enhance search results' relevance and diversity.

While dense embeddings provide superior semantic understanding, sparse embeddings remain widely used due to their interpretability, efficiency, and effectiveness in resource-constrained environments. Many modern IR systems employ hybrid models, combining sparse and dense representations for improved retrieval performance [1].

Quantization for Dense and Sparse Vectors
Quantization techniques are employed to reduce memory usage and improve computational efficiency during vector search. Product quantization (PQ) is used for dense vectors, which compresses vectors by dividing them into smaller subspaces, significantly enhancing search speed without compromising accuracy. Both Milvus and FAISS support PQ, allowing them to handle extensive collections of high-dimensional vectors effectively. In addition, sparse vectors, such as those used in some of the neural

models, are indexed using sparse inverted indexing methods in Milvus - a feature not supported on FAISS. These methods ensure that the many zero values in sparse data are efficiently managed, allowing for faster retrieval.

2.1.3. Monolingual NIR models

The NIR models discussed earlier—dense embeddings, ANN-based retrieval, and fine-grained ranking—are instantiated in advanced retrieval architectures such as SPLADE and ColBERT. These models exemplify how modern retrieval systems extend beyond classical lexical matching by leveraging deep learning to redefine data representation and similarity search.

A significant monolingual IR model is SPLADE (Sparse Lexical and Expansion model) [14]. SPLADE is an extension of sparse representations that incorporates neural contextualization while maintaining compatibility with classical IR indexing techniques. SPLADE addresses the limitations of traditional Bag-of-Words (BOW) models and dense embedding approaches by predicting the importance of terms in documents using logits from BERT's masked language modeling (MLM). The model generates sparse vectors for documents and queries, which can be efficiently indexed using inverted indexes—a technique reminiscent of traditional IR systems. At search time the encoded query is used by an inverted index to match terms between the document and the query [14].

ColBERT (Contextualized Late Interaction over BERT) exemplifies the dense embedding paradigm by structuring retrieval around fine-grained token-level embeddings [15]. Unlike other NIR techniquess, which ranks documents based on the entire document representations, ColBERT encodes each token in the query and document separately into dense vectors. This aligns with the token-level embeddings described earlier in NIR, where embeddings are not fixed but instead context-dependent. The pipeline begins by separately encoding the query and document using a BERT-based encoder. Each token is transformed into a high-dimensional vector, capturing its contextual meaning. These token-level vectors are then stored for documents in a pre-computed, indexed format, which allows for efficient reuse during retrieval. When a query is processed, ColBERT uses its MAXSim operator to compute the maximum similarity (via inner product) between each query token vector and all document token vectors. During the search, ColBERT performs an initial retrieval stage to identify a candidate set of potentially relevant documents using approximate nearest neighbor (ANN) techniques on the pre-computed embeddings. These candidate documents are then reranked using ColBERT's late interaction mechanism, which evaluates fine-grained token-level similarities between the query and document embeddings via inner product. This reranking step ensures that ColBERT refines the relevance ranking by focusing on token-level matches critical for precise retrieval.

2.2. Multilingual search in Information retrieval

2.2.1. Classical approaches

Classical approaches to multilingual search in IR primarily rely on lexical-based methods that adapt traditional IR techniques to handle multiple languages. These methods are typically designed to either create separate indexes for different languages or translate queries and documents to a common language before performing retrieval. While effective in structured multilingual environments, these classical approaches still struggle with semantic mismatches, translation errors, and resource constraints in low-resource languages.

Language-specific indexing entails creating separate indexes for each language present in a corpus of data [9]. Queries are processed in their original language and are then matched against the corresponding language index. In this method, separate indices are created for each language, with each index configured with language-specific analyzers tailored to the characteristics of that particular language. At search time, queries are directed to the appropriate language-specific index based on the identified language of the query, ensuring that the search leverages the correct linguistic processing for accurate retrieval.

Query or Document translation approaches typically involve translating into a target or pivot language of the documents collection using bilingual dictionairies or machine translations systems (which involves neural learning models to translate but not for retrieval).An example of this is by Fujii et al. [13]. In this paper, the authors address the challenge of retrieving English documents using Japanese queries.

They employ a method where Japanese queries are translated into English using a combination of dictionary-based translation and transliteration techniques. The translated queries are then used to perform retrieval on an English document collection utilizing traditional lexical matching methods.

2.2.2. Neural approaches

Unlike classical methods that rely on explicit translation or separate indexes, neural approaches directly encode multilingual text into a shared representation space. These methods can be categorized into multilingual embeddings and neural retrieval techniques.

Multilingual Pretrained Language Models (MPLMs) are common methods to generate embeddings for data in a multiple language and represent them in the same latent space. MPLMs learn cross-lingual representations to facilitate search and retrieval across different languages. Models such as mBERT, XLM, and XLM-R have significantly advanced cross-lingual understanding in IR tasks by encoding semantic relationships across languages:

- mBERT: A multilingual extension of BERT, mBERT is trained on a masked language modeling objective across 104 languages using a shared vocabulary. This training allows it to learn language-agnostic representations, supporting effective cross-lingual transfer tasks such as translation and retrieval [20].
- XLM: Building on mBERT, XLM introduces a translation language modeling (TLM) objective, which aligns representations between languages by training on parallel sentence pairs. This approach improves the model's performance on tasks requiring cross-lingual understanding, including retrieval and machine translation [17].
- XLM-R: A more advanced model, XLM-R is trained exclusively on monolingual corpora from a significantly larger and more diverse dataset. This enables it to achieve superior performance across multilingual tasks, particularly in low-resource languages, by generalizing better across diverse linguistic contexts [10].

Once text is embedded in a multilingual vector space, NIR systems use various retrieval methods; vector database enabled retrieval and dense vector search to find the most relevant documents.

- Jina-Colbert v2 [2]: Built on XLM-R, this ColBERT-style model extends multilingual retrieval without using language-specific adaptations or embeddings. Uses separate query/document encoding followed by late interaction matching via inner product similarity, improving precision on multilingual retrieval tasks [2].
- Splade-X [3]: Adapted for Cross-Language Information Retrieval (CLIR) from SPLADE, this model generates sparse vector representations for documents and queries while leveraging multilingual BERT (mBERT) . By expanding terms based on BERT's masked language modeling (MLM) predictions, SPLADE-X reduces noise and enhances retrieval efficiency.
- TranslateDistill [39]: Combines machine translation (MT) and knowledge distillation to train dense retrieval models for CLIR without direct translation during retrieval . Machine translation standardizes data, allowing documents in multiple languages to be mapped into a common representation space. A teacher-student distillation approach transfers knowledge from a larger teacher model to a lighter, faster student model optimized for retrieval.

Neural approaches to multilingual IR focus on learning cross-lingual representations and optimizing retrieval through vector search techniques. Multilingual Embeddings ensure text from different languages is represented in a unified vector space. Neural Retrieval Techniques (Jina-ColBERT-v2, SPLADE-X, TranslateDistill) utilize these embeddings and apply to enable scalable and accurate multilingual search and reduce reliance on explicit translations.

The effectiveness of these multilingual embeddings and retrieval techniques depends not only on their architecture but also on how they are trained. Different training strategies influence a model's ability to generalize across languages, directly impacting retrieval performance in multilingual settings. For instance, English Training (ET) trains models exclusively on English data but often lacks the linguistic diversity needed for robust cross-lingual performance. In contrast, Multilingual Translate Training (MTT) directly exposes models to the same data in multiple languages during training, improving their ability to handle diverse queries. MTT can be implemented using mixed-language batches (MTT-M) or

single-language batches (MTT-S). Research shows that MTT-M, which immerses the model in diverse linguistic contexts, generally outperforms MTT-S and ET, demonstrating the effectiveness of a truly multilingual training environment in enhancing retrieval performance across languages [18].

2.2.3. Training and fine-tuning

NIR models require both initial training and fine-tuning to achieve optimal performance in multilingual settings. Training typically involves learning general retrieval representations from large-scale datasets, while fine-tuning adapts pre-trained models to specific tasks or domains with data that closely aligns with the intended use case. In multilingual IR, fine-tuning is crucial for improving cross-lingual generalization and retrieval performance.

Negative sampling

To conduct training and fine-tuning, having positive and negative references for a query is essential because they help the model distinguish between relevant and irrelevant examples. This contrastive learning approach enables the model to weigh the importance of various features to determine relevance. In information retrieval, sampling negatives are often done using a baseline model such as BM25.

For negative sampling, the simplest approach is to take results beyond the top-K retrieved passages as negatives. For instance, negatives are sampled from positions K+1 onward, ignoring the highest-ranked results, which are more likely to overlap with the query in terms of lexical features. While effective, this method risks biasing the model against documents with high query-term overlap [4]. Another strategy focuses on sampling passages that match query tokens but do not contain the correct answer or are contextually dissimilar to the query. To identify these passages, cosine similarity between the query and potential negatives can be calculated, selecting the least similar passages as stronger negatives [29].

Negative sampling strategies tailored for multilingual settings include [44]:

- Mixed-Language Batches: By combining passages in different languages within the same batch, models are trained to rank cross-lingual pairs effectively. This approach has been particularly successful in training multilingual NIR models .
- Single-Language Batches: Alternatively, some strategies focus on sampling negatives from a single language per batch to isolate the effect of language-specific training. While less effective in generalizing across languages, this method can reduce computational overhead and memory requirements.
- Round Robin Strategies: This approach repeats the same query with passages in various languages, creating multilingual triples. While effective, memory limitations often constrain the number of queries that can be processed in a single batch.

One popular technique in training NIR models is the use of in-batch negatives [25], where negative examples for a query are sampled from the results retrieved for other queries in the same batch. This method involves running a batch of queries, retrieving results using BM25, and constructing a similarity matrix (e.g., via cosine similarity). The top result for a query serves as the positive example, while all other results are considered negatives.

Training NIR models

Training typically begins with large-scale unsupervised pair training using in-batch negatives, followed by smaller-scale triplet fine-tuning [2]. This two-step process significantly improves performance compared to training on triplet data alone. Additionally, incorporating synthetic translated data during training enhances the model's multilingual capabilities

- Single-Vector Models: Training typically begins with large-scale unsupervised pair training followed by smaller-scale triplet fine-tuning. This two-step process significantly improves performance compared to training on triplet data alone. Additionally, incorporating synthetic translated data during training enhances the model's multilingual capabilities [2].
- Multi-Vector Models: Approaches such as ColBERT use multiple smaller embeddings for tokens instead of a single large vector for queries and passages. These models benefit from techniques like cross-encoder distillation, self-mined hard negatives, and multilingual batching [2].

- Negative Sampling Techniques: Different strategies exist for obtaining negatives, such as BM25-based negatives, hard negatives, and in-batch negatives, each providing distinct advantages for model training [25].

Fine-Tuning NIR Models for Multilingual Retrieval
Fine-tuning is the process of adapting a pre-trained NIR model to a specific task or domain by continuing training on a smaller, task-specific dataset.

For multilingual settings, several approaches to fine-tuning have been explored:

- Language-Mixing Strategies: Mixed-language batching strategies, such as Multilingual Translate-Train (MTT), expose the model to documents and queries in multiple languages within the same batch. This method, referred to as MTT-M, has been shown to outperform single-language batching (MTT-S) in terms of cross-lingual retrieval performance by reducing language bias and improving the alignment of multilingual semantic spaces [18].
- Data Language Composition: Some models construct training triples exclusively in a single language, avoiding cross-language pairs within individual triples. In contrast, other approaches, such as "mix passages" and "mix entries," introduce diversity by allowing passages in a triple to be in different languages, which directly trains the model to handle cross-lingual ranking [18][2].
- Multilingual Datasets: High-quality multilingual training datasets can be generated via machine translation or human annotation. While machine-translated datasets provide scale, human-generated datasets tend to deliver higher quality and contextual relevance [7].

2.2.4. Evaluating fine-tuned models
Research [19] indicates that models fine-tuned on specific data formats perform optimally when evaluated in the same format. Here format refers to the structure or type of data present in the corpus. In the context of Europeana's metadata this would encompass the document enrichments or translations.

For instance, a study on instruction tuning (a technique for fine-tuning LLMS on a labeled dataset of instructional prompts and corresponding outputs) found that maintaining format consistency between training and evaluation data is crucial for optimal performance. Additional literature [42] emphasizes the importance of data quality and consistency in training datasets.

2.3. Multilingual neural models
By combining the strengths of multilingual embeddings, semantic retrieval techniques, and advanced training strategies, neural systems have made significant strides in providing multilingual search experiences. Some state-of-the-art multilingual neural models include: Jina-ColBERT-v2, BGE-M3, Multilingual SBERT, SPLADE-X, and TranslateDistill,

Jina-Colbert-v2
Jina-ColBERT-v2 is an advanced model designed for multilingual and code-switched information retrieval. It builds on XLM-Roberta, extending its capabilities without introducing language-specific adaptations or embeddings [2]. This design allows Jina-ColBERT-v2 to handle code-switching seamlessly, as it does not modify its behavior based on language detection during the embedding phase. The model introduces key enhancements, including flash attention and rotary positional embeddings (RoPE). Flash attention provides a faster and more memory-efficient mechanism for identifying the most relevant parts of a sentence or document when processing a specific word, improving performance on large-scale datasets. RoPE improves the model's understanding of word order in sentences, allows it to handle longer contexts effectively [2]. Like the original ColBERT framework, Jina-ColBERT-v2 performs separate encoding of queries and documents, followed by a late interaction step to calculate the similarity between encoded query vectors and document vectors. This efficient architecture, combined with its enhancements, makes Jina-ColBERT-v2 highly suitable for mixed-language retrieval scenarios without requiring explicit language adapters or modifications[2]. This variant of colbert also has the added benefit of having much larger input sequence length of 8192 tokens which is well suited for the document sizes Europeana has. Jina-ColBERT-v2 utilizes ColBERTv2's centroid-based encoding, which represents each token embedding vector (v) as the sum of its nearest centroid (C_t) and a quantized

residual vector ($\tilde{r}$). Specifically, the vector is encoded as:

$$v \approx \tilde{v} = C_t + \tilde{r}$$

where C_t is the closest centroid from a predefined set C, and $\tilde{r}$ is the quantized approximation of the residual vector:

$$r = v - C_t$$

This encoding significantly reduces storage requirements by storing only the index of C_t and the quantized residual $\tilde{r}$, with $\tilde{r}$ being compressed into n-bit representations per dimension, ensuring compactness while preserving retrieval performance.

BGE-M3
BGE-M3 [1], short for Multi-Linguality, Multi-Functionality, Multi-Granularity Embedding, is a highly versatile model tailored for multilingual information retrieval across over 100 languages. It stands out by supporting three retrieval functionalities—dense retrieval, sparse retrieval, and multi-vector retrieval—while accommodating inputs ranging from short queries to long documents with an input sequence length of 8192 tokens. - making it well-suited for Europeanas data. The model leverages a novel self-knowledge distillation approach, integrating relevance signals from different retrieval methods to enhance training quality. BGE-M3 is trained on a diverse dataset comprising unsupervised multilingual corpora, fine-tuned labeled data, and synthetic long-document retrieval examples, resulting in a robust embedding space for both multilingual and monolingual retrieval tasks. Its hybrid retrieval framework combines dense and sparse vector scores for re-ranking, enabling superior performance across multilingual benchmarks such as MIRACL and MKQA. BGE-M3's comprehensive design makes it a state-of-the-art choice for scalable and high-quality multilingual retrieval systems.

SBERT: Multilingual DistilUSE
Sentence Embeddings encompass various models that map sentences and paragraphs to an n-dimensional space and are used for clustering and semantic search. Sentence-BERT (SBERT) models [32], specifically the Multilingual DistilUSE model [31][32], is a notable state of the art model that accomplishes this task. SBERT adapts the BERT architecture, employing siamese and triplet network structures to generate semantically meaningful sentence embeddings. This design enables the computation of cosine similarity between sentence embeddings, significantly enhancing the efficiency of tasks like semantic textual similarity and paraphrase identification [31]. The Multilingual DistilUSE model extends these using DistilBERT to support multiple languages via a smaller and cheaper transformer model. It maps sentences from various languages into a shared semantic space, facilitating cross-lingual retrieval and semantic similarity tasks. Multilingual DistilUSE is trained on a diverse, multilingual corpus, enabling it to capture semantic nuances across different languages and perform effectively in multilingual information retrieval scenarios [31]. This SBERT model has a much smaller input sequence length of 128 as it is primarily meant for sentence-level embeddings.

SPLADE-X
SPLADE-X is a model adapted for Cross-Language Information Retrieval (CLIR) based on the SPLADE architecture [3]. SPLADE-X leverages multilingual BERT (mBERT) to handle documents and queries in different languages. It generates sparse vector representations for documents and queries, focusing on the most relevant terms to reduce noise and enhance retrieval efficiency [3].

Translate-Distill
Translate-Distill combines machine translation and knowledge distillation to train dense retrieval models specifically for CLIR without direct translation during the retrieval process [39]. The approach starts with machine translation to standardize the data, followed by knowledge distillation from a teacher model to a more efficient student model [39]. Optimized for faster retrieval, this student model captures the essence of the query-document relationships learned from the teacher model, making it suitable for scalable and efficient multilingual IR tasks [39].

Model summary
The following tables contain a summary of the models discussed.

Model	Dimensions	Languages	Multilingual Approach	Level
Jina-ColBERT-v2	Multivector	100+	XLM-R based, late interaction	Token-level (Late interaction)
BGE-M3	1024	100+	Hybrid: dense, sparse, multi-vector	Sentence & Document-level - Multi grandular
Multilingual SBERT	512	50+	Siamese/Triplet BERT	Sentence-level
SPLADE-X		10+	Sparse vector expansion with mBERT	Sentence-level
TranslateDistill			MT + knowledge distillation	Document-level

Table 2.1: General Model Properties

Model	Datasets Trained On	Software	Seq Length
Jina-ColBERT-v2	Mr-Tydi, MIRACL, mMARCO	FAISS	8192
BGE-M3	MIRACL, MKQA	Milvus (model built into milvus)	8192
Multilingual SBERT	Stanford Natural Language Inference (SNLI) translated with google translate	Sentence-Transformers, Milvus	128
SPLADE-X	MIRACL		
TranslateDistill	NeuMarco mMarco		180 token sliding window

Table 2.2: Training and Retrieval Details

Splade-X and TranslateDistill do not have any models available currently so we were unable to get complete information for them.

2.4. Code switching in Multilingual-IR

Code-switching refers to the phenomenon where speakers alternate between two or more languages within a single conversation. In the context of text-based information retrieval, code-switching is the case where queries or documents include elements from multiple languages within a single instance [21]. This presents unique challenges for traditional Information Retrieval (IR) systems, as these multilingual models need to generalize multiple languages within a single block of text. In the context of this project, code-switching is particularly relevant due to the multilingual nature of Europeana's dataset due to translations and enrichments.

Multilingual NIR models, such as mBERT, XLM-R, or BGE, offer a promising approach to handle code-switched data effectively. These models are trained on multilingual corpora and can represent text in a shared semantic space, making them inherently capable of processing mixed-language inputs. The authors of XLM-R state that they "do not use language-specific embeddings, which allows the model to better deal with code-switching."[10].

However, code-switching introduces additional complexity, as the models need to understand not only different languages but also the relationship between languages within a single input. Research indicates that code-switching can challenge multilingual models, as they may struggle with the dynamic nature of language switching [43]. Moreover, the scarcity of code-switched data for training further complicates the development of models capable of handling such inputs [45]. Addressing these challenges is crucial for enhancing the performance of multilingual IR systems in processing code-switched data.

2.5. Applications of MLIR

Cross-lingual and Multilingual Information Retrieval approaches have been increasingly applied in diverse domains, offering innovative solutions to bridge language barriers in information retrieval.

2.5.1. Cultural Heritage

The Nasjonalmuseet's semantic search prototype [26] represents a sophisticated application of CL/ML-IR in the cultural heritage sector. This system uses OpenAI's GPT-4 Vision API to extract rich, descriptive texts from digital images of artworks, capturing not only the pictorial content but also the thematic and emotional nuances of the art pieces. These descriptions are transformed into numerical embeddings using OpenAI's Embeddings API, enabling semantic search capabilities that transcend traditional keyword-based retrieval. By utilizing MongoDB Atlas's Vector Search with a K-nearest neighbors (KNN) algorithm, the system can efficiently match user queries—entered in any language—with the most relevant artworks, thus making the museum's collection accessible to a global audience.

Another significant application of CLIR is demonstrated in "A Cross-Language Approach to Historical Document Retrieval," [16] where researchers tackle retrieving documents in the cultural heritage domain similar to Europeana's use case, precisely, historic Dutch documents using modern Dutch queries. The system automatically constructs translation resources for the landmark language by comparing phonetic sequence similarity, consonant and vowel sequence frequencies, and character n-gram frequencies between historic and modern Dutch. This approach effectively bridges the gap between the archaic language of historical documents and contemporary language, enabling more accurate retrieval.

2.5.2. Other domains

In the legal domain, the paper "Translating Justice: A Cross-Lingual Information Retrieval System for Maltese Case Law Documents" [5] details a CLIR system designed for Maltese legal case documents. This system addresses the challenges posed by the Maltese language's status as a low-resource language in Natural Language Processing (NLP) by employing Neural Machine Translation (NMT) to translate Maltese legal documents into English. The system enables dual-language querying, allowing users to search in Maltese and English, and presents results in both languages.

These applications demonstrate the broad potential of CL/ML-IR systems across different domains, from enhancing cultural heritage exploration and legal research to facilitating the retrieval of historic documents. Each case underscores the value of leveraging cross-lingual capabilities to make information more accessible and relevant in multilingual contexts, thereby overcoming linguistic barriers and expanding the reach of digital content.

3

Methodology

3.1. Systems

System configurations

The selection of models and retrieval strategies for this investigation was guided by multiple considerations related to Europeana's multilingual metadata. This required examining key decisions in model selection, fine-tuning, query handling, and document representation, particularly in relation to existing practices at Europeana.

Europeana currently relies on BM25 in Solr for retrieval, which does not leverage neural embeddings. To explore potential improvements, we selected various retrieval models to assess their retrieval effectiveness and multilingual support.

While some models might perform well out-of-the-box, others might require fine-tuning. The purpose of fine-tuning is to train the models not only on Europeanas domain but to also teach the model about Europeanas multilingual code-switched data. Helping the models better handle queries and documents that mix languages, ensuring improved retrieval performance across Europeana's diverse multilingual collections. .

In their pilot project Europeana translates non-English queries into English for BM25 retrieval. We will examine whether this approach is necessary for neural models or if multilingual embeddings could retrieve relevant documents directly in multiple languages.

For documents, Europeana's metadata includes multiple layers of augmentation on top of the provided data, such as enrichment and translations. In the current Solr-based system, these augmentations are indexed alongside the original metadata. We investigated whether neural retrieval models benefit from indexing only the provided metadata or if using enriched and translated content leads to better retrieval performance. This decision impacts model efficiency, storage requirements, and retrieval accuracy.

Overall, these are the considerations we take into account when deciding what models to use, their configurations, and how to structure the dataset we need to experiment with.

3.1.1. BM25 in SolR

BM25 is used as a baseline model for evaluation. This method will provide a benchmark against which the performance of NIR models can be compared. Using this model provides a robust baseline for comparison, as it mirrors Europeana's production environment. The performance of NIR model will be evaluated against this standard to assess the potential benefits or limits.

3.1.2. Jina Colbert V2

ColBERT uses token-level representations. This token-level granularity is particularly beneficial for Europeana's diverse and metadata-rich documents, where different sections of a document (e.g., provided metadata, enriched metadata, translations) may be relevant to different queries. Instead of

compressing all document information into a single dense vector, ColBERT retains finer distinctions between words and phrases, improving retrieval accuracy in heterogeneous metadata scenarios.

ColBERT's multilingual embedding space aligns well with Europeana's multilingual queries and documents. Since Jina-ColBERT-V2 is based on XLM-R, a model trained on over 100 languages, it provides strong cross-lingual retrieval capabilities without requiring explicit query translation. This contrasts with BM25, which relies on translated queries to work in multilingual settings. Using ColBERT's token-level embeddings, the system can possibly retrieve relevant documents in the original language of the query, making it a more effective solution for Europeana's multilingual and code-switched retrieval needs.

The Jina-ColBERT-V2 demonstrates strong multilingual retrieval performance across benchmarks like BEIR, LoTTE, MIRACL, and mMARCO [2], outperforming BM25 and zero-shot mDPR while maintaining competitive results against fine-tuned models. Thus its possible that ColBERT performs well out-of-the-box, but fine-tuning can help adapt it to Europeana's domain-specific metadata.

Jina Colbert V2 was implemented using Stanford-FutureData's colbert library which has inbuilt indexing, search, and training functionality. This library will utilise Jina-AI's Colbert V2 multilingual checkpoint available through huggingface. This checkpoint is also compatible with Stanford-Futuredata's library. The library's indexing uses FAISS-GPU to store the computed vectors in the users' local directory.

Although Milvus was the primary consideration as the vector database for this investigation, we use FAISS due to its integration within the Stanford-FutureData library. Furthermore, Milvus is not used for Colbert because it is not currently designed to handle ColBERT's multivector embedding structure, which involves token-level embeddings rather than a single dense vector per document.

To facilitate indexing, the data was prepared in a TSV format, where each line contains a numeric ID and its corresponding passage text. This format is required by the ColBERT indexer to map documents to their token-level embeddings efficiently. Since Europeana IDs are alphanumeric, a mapping was created to convert these IDs into numeric primary keys, ensuring compatibility with the ColBERT framework. The indexing process leverages FAISS-GPU to store the computed vectors, allowing for scalable and efficient similarity searches.

The indexing configuration was carefully tailored to optimize performance. The nbits=1 parameter defines the residual vectors in Colberts compression scheme. This setting ensures that the storage footprint is significantly reduced without compromising retrieval accuracy. The kmeans_niters=2 parameter controls the number of iterations for k-means clustering during the indexing process, chosen to reduce computation time while maintaining adequate cluster quality for effective token-level retrieval. These decisions were made to align with the high-dimensional, large-scale nature of the Europeana dataset, ensuring efficient storage and retrieval for subsequent querying. Finally, we used product quantization (PQ) because it reduces memory usage and improves computational efficiency during similarity searches without compromising retrieval accuracy.

The querying process for ColBERT utilizes the Searcher class from Stanford-FutureData's ColBERT library, enabling efficient retrieval with support for both original and translated queries.

3.1.3. BGE-M3 - Hybrid model

Along with a fine-grained multi-vector dense embedding model, we also chose to test a hybrid dense-sparse embedding model. Europeana's metadata consists of structured text fields, enriched descriptions, and multilingual metadata, possibly making a purely dense or purely sparse approach insufficient.

While BM25-based lexical search excels at exact term matching, it lacks semantic understanding. Conversely, dense embeddings provide strong semantic matching but can miss exact keyword-based retrieval cues, especially for proper nouns, entity terms, and specific metadata fields. By combining dense and sparse representations, the BGE-M3 hybrid model aims to leverage the strengths of both approaches.

BGE-M3 was chosen because it is designed for retrieval and provides both dense and sparse embeddings natively. A Dense vector (1024 dimensions) captures contextual meaning in multilingual metadata, while a sparse vector enhances retrieval of specific terms missing in dense representations.

Results from the dense and sparse searches are merged using Reciprocal Rank Fusion (RRF), which combines rankings from the two modalities by assigning higher scores to top-ranked results from each list. RRF is a rank aggregation method that combines the ranked lists from multiple searches by assigning higher weights to documents that rank higher in either list. Specifically, the RRF score for a document d is calculated as:

$RRF(d) = \sum_{i=1}^{n} \frac{1}{k+rank_i(d)}$

where n is the number of input ranked lists, $rank_i(d)$ is the rank of document d in the i-th ranked list (or a large value if d is not present in that list). This ensures a balanced integration of semantic and lexical relevance. The implementation supports flexible querying, including handling translated and original queries, with RRF dynamically merging their results to improve retrieval accuracy for multilingual and heterogeneous datasets.

BGE-M3 is implemented using Milvus as the vector storage solution. Milvus is used for storage because it supports hybrid search between sparse and dense vectors, something which FAISS does not natively support. Furthermore, the BGE-M3 model is supported by and built into milvus, making its implementation simpler and suitable for Europeana's high-dimensional vector data.

The schema for the index is designed to accommodate the specific requirements of Europeana's dataset. The primary key is defined as the Europeana ID, which can be an alphanumeric string. This ensures compatibility with Europeana's unique identifiers. The schema includes a dense vector of size 1024, generated by BGE-M3, to represent the semantic embedding of the document, capturing its contextual meaning, and a sparse lexical vector to facilitate hybrid retrieval, combining the precision of lexical matching with the semantic richness of dense embeddings. The sparse vector does not require a predefined dimensionality, reflecting its adaptable structure for encoding keyword-based information. The actual document text is stored as a separate field, allowing for direct access during retrieval and evaluation.

For indexing, product quantization (PQ) is employed for the dense vector using the IVF_PQ index type, with inner product (IP) as the similarity metric. PQ is chosen because it reduces memory usage and enhancing retrieval speed without compromising too much on retrieval accuracy. The sparse vector is indexed using the SPARSE_INVERTED_INDEX, which enables efficient keyword-based retrieval. This hybrid schema and indexing strategy provide a robust foundation for handling Europeana's multilingual and heterogeneous data, offering scalability, high retrieval performance, and optimal resource utilization.

3.1.4. SBERT: Multilingual DistilUSE

The previous models (ColBERT and BGE-M3) are quite resource-intensive, requiring large embeddings to handle complex token-level or hybrid retrieval. While they maximize retrieval effectiveness, they also increase computational cost and storage requirements.

The SBERT model, specifically Multilingual DistilUSE, was chosen as a much lighter model allowing us to test whether a smaller dense embedding models can still achieve competitive retrieval performance. This model is well-suited for sentence and document-level retrieval, which aims to match queries with semantically similar texts rather than relying on exact term matches. SBERT also supports multilingual retrieval and can be fine-tuned for specific datasets, which allows it to adapt to Europeana's multilingual metadata challenges.

SBERT supports multilingual retrieval, meaning queries in different languages can be matched to relevant documents without explicit translation. If SBERT performs sufficiently well, it presents a computationally cheaper and possibly faster alternative, making neural retrieval more feasible and cost effective for Europeana.

The SBERT model is implemented using Milvus as well. The model is made available through the sentence transformers library via Huggingface. The SBERT indexing pipeline is implemented using Milvus as the vector storage system, with embeddings generated by the lightweight, multilingual distiluse-base-multilingual-cased-v2 model from SentenceTransformers. The schema is designed to include a alphanumeric primary key for unique identifiers, the document text, and a dense vector of size 512 for semantic representations.

As previously mentioned, this model can only process input of up to 128 tokens. To handle longer texts and improve retrieval granularity, documents are chunked into smaller segments using a custom chunking strategy that balances context preservation and token limits. We split each passage into chunks of at most 512 characters. According to research by OpenAI, the approximate conversion rate of 1 token 4 characters [1] means that a 512-character chunk generally corresponds to around 128 tokens. Following this estimation we can see that the tokenized input remains within the model's limits, minimizing truncation while maintaining sufficient context for effective retrieval. Each chunk is assigned a unique identifier based on the original document ID, allowing precise mapping during retrieval. While this estimation provides a useful guideline, it is important to note that the exact token count may vary depending on the tokenizer used. Different tokenization algorithms and SentencePiece, segment text differently based on vocabulary size and tokenization rules.

The SBERT querying process leverages Milvus for dense vector search, incorporating chunk-level retrieval to handle long documents effectively. Queries are embedded using the SBERT model and queried against the dense_vector field in Milvus using inner product (IP) similarity. To account for chunked documents, results are retrieved for individual chunks and aggregated at the document level using averageP method [8] which gives a score to a document based on its average chunk score from the number of chunks that found for that document based on the top k results.

Overall, this approach ensures scalable, and semantically rich retrieval, aligning with Europeana's dataset requirements.

3.1.5. System Identifiers

To systematically reference different retrieval system configurations in this investigation, we define a notation using the syntax:

$$\langle \text{model} \rangle \langle \text{finetune} \rangle - \langle \text{qaug} \rangle - \langle \text{daug} \rangle$$

where:

- ⟨model⟩ represents the retrieval model:
  - **B** – BM25 (Solr-based retrieval)
  - **C** – ColBERT (Jina-ColBERT V2)
  - **H** – Hybrid (BGE-M3 dense + sparse retrieval)
  - **S** – SBERT (Multilingual DistilUSE)
- ⟨finetune⟩ represents the fine-tuning condition:
  - **Z** – Zero-shot (pretrained model, no fine-tuning)
  - **F** – Fine-tuned on Europeana-specific data
  - (Empty for BM25, as fine-tuning does not apply)
- ⟨qaug⟩ represents query augmentation:
  - **O** – Original queries only
  - **OT** – Original + Translated queries
- ⟨daug⟩ represents document augmentation:
  - **P** – Provided metadata only
  - **PE** – Provided + Enriched metadata
  - **PT** – Provided + Translated metadata
  - **PET** – Provided + Enriched + Translated metadata

For example:

[1] https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-themh$_3$4f2b50bab

- **CZ-O-P** – ColBERT, zero-shot, original queries only, provided metadata.
- **HF-OT-PET** – Hybrid BGE-M3, fine-tuned, original and translated queries, provided, enriched, and translated metadata.
- **SZ-O-PE** – SBERT, zero-shot, original queries only, provided and enriched metadata.
- **B-O-P** – BM25, original queries only, provided metadata.

We explicitly outline all 56 systems in table 3.1

3.2. Dataset

For this investigation, Europeana did not provide any data for us to use for our experiments. We were provided access to their API and an FTP server from which we could download their raw xml-structured data, however, a dataset suitable for NIR was not available.

Therefore, a crucial task for this thesis was to design a methodology for constructing a dataset suitable for NIR. This involved extracting and structuring documents and queries, ensuring multilingual consistency, and preparing the dataset for indexing, training, and querying. Developing this methodology was essential not only for conducting our experiments but also for establishing a foundation for future NIR research within Europeana.

For this investigation, we utilize need two essential datasets, from which we can gather all the necessary information for the follow stages of the investigation, namely the Document and Click dataset. The Document dataset comprised of Europeana documents for indexing and retrieval with the selected retrieval models. This dataset comprises metadata-rich records that serve as the foundation for search and ranking experiments. The click dataset comprised user queries and click interactions. This includes query logs, clicked document records, and supplementary metadata. The Click Dataset is used for fine-tuning retrieval models and evaluating search performance, offering insight into real-world user interactions. And the queries found from the click data were used to query the indexed document data and conduct the experiments.

These datasets serve distinct but complementary roles: the Document Dataset provides the retrieval corpus, which is indexed using the models, while the Click Dataset supplies the queries we use for retrieval and query-document relational data which provides us with relevance signals for training and evaluation.

3.2.1. Documents

Data collection methodology

As outlined earlier the document dataset did not exist and was not provided by Europeana. Therefore, when we are building the dataset we have to consider the factors which we will be analysing for this investigation, which we outlined in section 3.1. So considering the document specifciations outlined (the enrichments and translations), we need to build a dataset that can allow us to isolate these components so we can observe the effects of NIR retrieval with and without them.

The document in Europeana primarily consists of provided documents, which are metadata records as supplied by data providers. This dataset serves as the core collection for indexing and retrieval across all experiments, forming the foundation upon which search models operate.

A crucial aspect to recognize is that enrichments and translations are not inherent properties of the dataset but rather a design choice applied once the documents are given by the provider. Europeana's data processing pipeline enriches all of the data they recieve adding entity related data from external knowledge sources such as Wikipedia, while translation efforts have been applied to a subset of the collection to facilitate multilingual access. However, these augmentations do not alter the fundamental nature of the Provided dataset. Instead, they act as additions that Europeana may choose to incorporate.

By maintaining a clear distinction between the raw dataset and the augmentations applied to it, we ensure consistency across our evaluation. This approach allows for a controlled investigation into the impact of different document augmentations on retrieval effectiveness without conflating them with the core dataset itself.

Model	Query Augmentation	Document Augmentation	Fine-tuning
BM25	O	O	N/A
BM25	O	O + E	N/A
BM25	O	O + T	N/A
BM25	O	O + E + T	N/A
BM25	O + T	O	N/A
BM25	O + T	O + T	N/A
BM25	O + T	O + E	N/A
BM25	O + T	O + E + T	N/A
Colbert	O	O	Zeroshot
Colbert	O	O + E	Zeroshot
Colbert	O	O + T	Zeroshot
Colbert	O	O + E + T	Zeroshot
Colbert	O + T	O	Zeroshot
Colbert	O + T	O + T	Zeroshot
Colbert	O + T	O + E	Zeroshot
Colbert	O + T	O + E + T	Zeroshot
Colbert	O	O	O
Colbert	O	O + E	O + E
Colbert	O	O + T	O + T
Colbert	O	O + E + T	O + E + T
Colbert	O + T	O	O
Colbert	O + T	O + T	O + T
Colbert	O + T	O + E	O + E
Colbert	O + T	O + E + T	O + E + T
Hybrid	O	O	Zeroshot
Hybrid	O	O + E	Zeroshot
Hybrid	O	O + T	Zeroshot
Hybrid	O	O + E + T	Zeroshot
Hybrid	O + T	O	Zeroshot
Hybrid	O + T	O + T	Zeroshot
Hybrid	O + T	O + E	Zeroshot
Hybrid	O + T	O + E + T	Zeroshot
Hybrid	O	O	O
Hybrid	O	O + E	O + E
Hybrid	O	O + T	O + T
Hybrid	O	O + E + T	O + E + T
Hybrid	O + T	O	O
Hybrid	O + T	O + T	O + T
Hybrid	O + T	O + E	O + E
Hybrid	O + T	O + E + T	O + E + T
SBERT	O	O	Zeroshot
SBERT	O	O + E	Zeroshot
SBERT	O	O + T	Zeroshot
SBERT	O	O + E + T	Zeroshot
SBERT	O + T	O	Zeroshot
SBERT	O + T	O + T	Zeroshot
SBERT	O + T	O + E	Zeroshot
SBERT	O + T	O + E + T	Zeroshot
SBERT	O	O	O
SBERT	O	O + E	O + E
SBERT	O	O + T	O + T
SBERT	O	O + E + T	O + E + T
SBERT	O + T	O	O
SBERT	O + T	O + T	O + T
SBERT	O + T	O + E	O + E
SBERT	O + T	O + E + T	O + E + T

Table 3.1: Model configurations for query augmentation, document augmentation, and fine-tuning.

It is important to note that these processes are not uniformly applied across Europeana's entire dataset. Not all documents undergo successful translation. Due to this variability we decided to look into the subset of Europeanas data which was translated successfully. While enrichments are always applied to Europeanas data.

Thus, for the document dataset, we decided to create 4 iterations each with different augmentations allowing us to isolate for and determine the impact of each. Following our system identifiers, the four datasets we created are:

- Provided only (P): The original data given by the providers
- Provided and Enriched (PE): Original data with Enrichments
- Provided and Translated (PT): Original data with Translations
- Provided, Enriched, and Translated (PET): Original data with both enrichments and translations.

These iterations will comprise of the same data just with varying degrees of augmentation, which simply captures additional information for the data. This ensures that any observed differences in retrieval performance can be attributed to the presence or absence of enrichments and translations.

Europeana's complete dataset contains over 60 million total records, represented in 40 different languages from around Europe. The collection exhibits significant diversity in metadata quality, with items categorized into content tiers (0 to 4) and metadata tiers (A to C) based on the level of detail provided. These classifications help standardize and distinguish the richness of information available for each item. The quality and completeness of translations also vary. Therefore, to create a dataset suitable for our retrieval experiments, we chose to focus on the subset of documents which were translated successfully and that had a content tier of 1 or higher (which is what Europeana does in production).

According to Europeanas logs from their translation process, 40,783,067 documents were successfully translated. These documents also had the enrichment process done on them; as with all data that Europeana retrieves. Through these logs we were able to identify the document ids of these 40 million documents and the ids of the datasets where they are stored.

Given that we have 40 million documents which were translated and enriched we clearly could not use all of them to conduct experiments with as it would be infeasible to store and run. Therefore, we decided to make some decisions pertaining to the number of documents and number of languages.

Regarding the documents, we decided to stratify by each dataset to take 10% of the documents in that dataset. This would result in a much smaller representative set but would still be more manageable to use than all 40 million documents. However, we also had to be careful about the which documents we would take as we want a representative dataset, for our experiments, which closely resembles the multilingualism of Europeanas translated data (of 40 million documents).

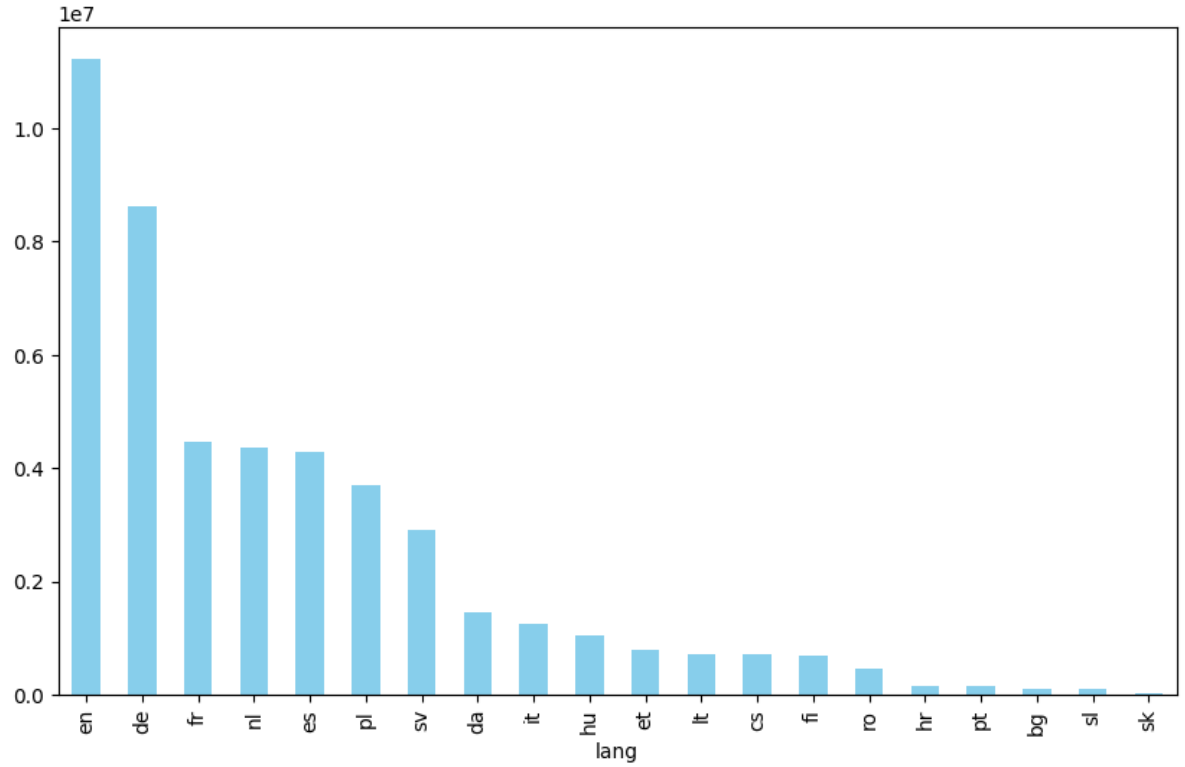


Figure 3.1: Language distribution of the full datasets

We analyzed the language distribution of documents across these translated documents and we chose the 20 most prominent languages with sufficient representation, excluding underrepresented languages such as Maltese. These languages include: English, Bulgarian, Czech, Danish, German, Spanish, Estonian, Finnish, French, Croatian, Hungarian, Italian, Lithuanian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovenian, and Swedish. As shown in figure 3.1, which illustrates the language distribution of the full dataset. We have the highest representation for English documents followed by other 'major' European languages such as German, French, Dutch, and Spanish. For all of these documents we have their document ids and the ids of the datasets where they are stored.

Overall this process would provide us with a representative set of data which resembles the multilingualism of the original set, contains documents which allow us to isolate for augmenations to conduct our experiments, and is of a manageable size for our experimentation with NIR. Allowing our experiments to appropriately give insights which can be extrapolated to the larger dataset.

Data collection process - technical aspects

The process of constructing these dataset required significant effort to collect, filter, and sample in a manner that aligns with both Europeana's multilingual scope and the practical constraints of our experimentation. Europeana's collection is structured into datasets based on different data providers, with each dataset containing records from a specific institution or collection. Accessing these records required downloading them from an FTP server, where each dataset is stored under a codified name corresponding to its provider. Since the dataset is multilingual and spans over multiple languages, it was also essential to ensure that our sample was representative of Europeana's actual language distribution.

This process involved several key steps: first, we downloaded metadata records per dataset from the FTP server. The raw data downloaded from the server is shown in figure 1.1. Next, we filtered for documents that had been successfully translated, using logs provided by Europeana that indicated which records had undergone translation successfully. Additionally, we restricted our selection to documents which were part of our set of 20 and documents with a content tier of 1 or higher, aligning with the filtering criteria used in Europeana's production search system. Finally, to manage dataset size

while preserving representativeness, we randomly sampled 10% of the available data, ensuring that the language distribution remained consistent with the full dataset.

For each document, we retained technical metadata such as the Europeana ID, content tier, metadata tier, and document type (e.g., image, painting, manuscript). We also extracted all metadata fields that map to Solr-BM25's "text" field, which includes descriptive information such as provider, creator, title, description, language, publisher, and current location. These fields contain rich textual content that is integral to document ranking and retrieval, ensuring that our dataset aligns as closely as possible with the data used in Europeana's production system. In the end each document which we downloaded has the Europeana id and the 'text' metadata.

To create the different dataset iterations (P, PE, PT, PET), we needed to separate the provided, enriched, and translated sections based on specific identifiers in the original XML structure. These identifiers allowed us to distinguish between the provided and the different augmentation levels applied to the metadata:

- <ore:Proxy rdf:about="http://data.europeana.eu/proxy/provider. . . > for the provided data
- <ore:Proxy rdf:about="http://data.europeana.eu/proxy/europeana> for the enriched data
- <ore:Proxy rdf:about="http://data.europeana.eu/proxy/europeana xml:lang="en"> for the translated data.

By leveraging these identifiers, we systematically extracted the relevant sections from each document and assigned them to their respective dataset iterations. This approach ensured that each dataset variant was constructed with the appropriate level of augmentation while maintaining a consistent document structure for comparative analysis

After filtering, downloading, and structuring, the resulting representative dataset maintained a language distribution that closely reflects Europeana's actual multilingual dataset. Figure 3.2 shows the language distribution of the downloaded dataset.

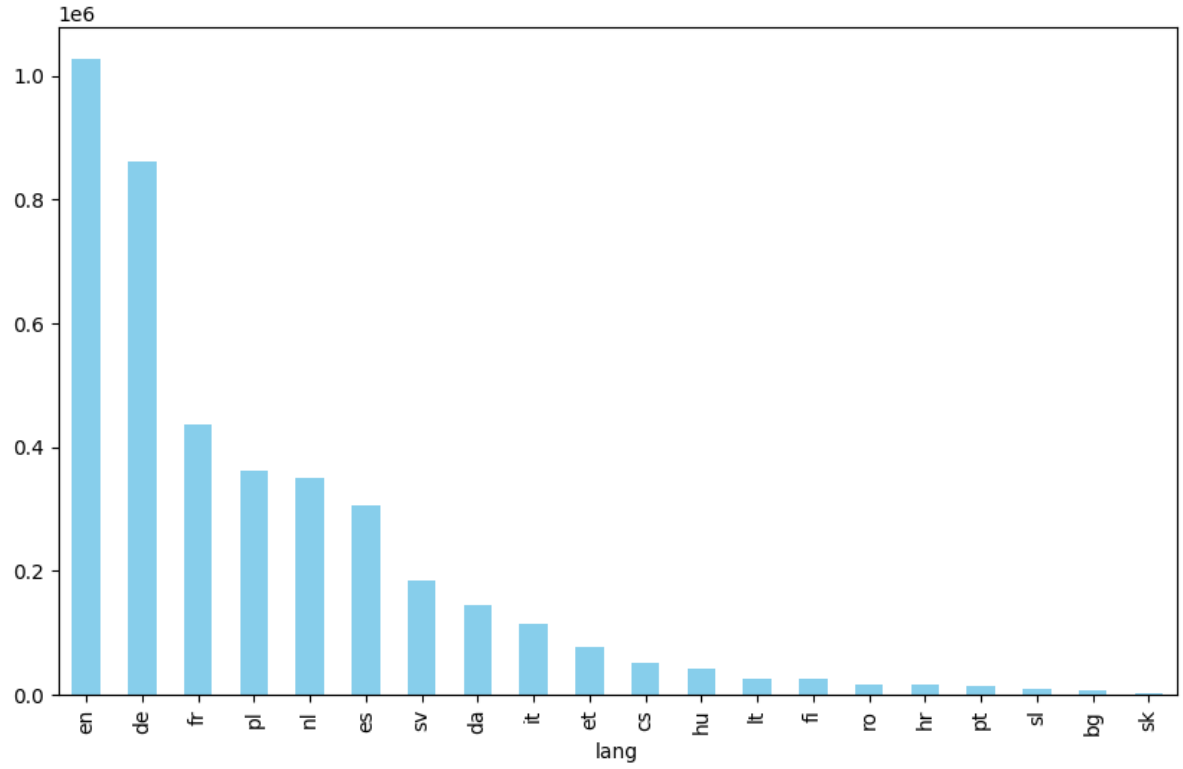


Figure 3.2: Language distribution of the representative datasets

The final dataset statistics are summarized in Table 3.2, comparing the original dataset size with the sampled dataset.

Original dataset count	1,488
New dataset count	1,290
Percentage of datasets after sampling	86.69%
Number of documents in original dataset	47,243,909
Number of documents in new dataset	4,072,603

Table 3.2: Comparison of Original and Sampled Dataset

By structuring the dataset in this way, we ensure that all retrieval models operate on the same underlying document collection, making it possible to conduct fair comparisons between different augmentation strategies. This setup allows us to investigate the impact of different augmentation strategies while keeping the underlying dataset constant, ensuring that results reflect the true effects of document enrichment's and translations rather than differences in the data itself. In the end our representative dataset has around 4 million documents from 1290 datasets. The difference from the original dataset count may be due to the fact that these datasets did not successfully translate the data.

Document representation

```
<add>
        <field name="europeana_id">/39/DDU</field>
        <field name="timestamp_update">2019-03-20T10:12:38.352Z</field>
        <field name="proxy_edm_type">TEXT</field>
        <field name="contentTier">3</field>
        <field name="metadataTier">A</field>
        <provided_data>
                <field name="provider_aggregation_edm_dataProvider">
                        <value lang="en">Opera Institute of the Italian Vocabulary</value>
                        <value lang="it">Istituto Opera del Vocabolario Italiano</value>
                </field>
                <field name="provider_aggregation_edm_provider">
                        <value lang="en">CulturaItalia</value>
                        <value lang="it">CulturaItalia</value>
                </field>
                <field name="proxy_dc_subject">
                        <value lang="en">Critical editions</value>
                        <value lang="it">Edizioni critiche</value>
                </field>
                <field name="proxy_dc_title">
                        <value>Sovrana ballata piacente</value>
                </field>
        </provided_data>
        <enriched_data>
                <field name="proxy_dc_title">
                        <value lang="it">Sovrana ballata piacente</value>
                </field>
        </enriched_data>
        <translated_data>
                <field name="proxy_dc_title">
                        <value lang="en">Great nice ballad</value>
                </field>
        </translated_data>
</add>
```

Figure 3.3: Example of Processed Europeana XML metadata

```
{
   "id":"/39/DDU",
   "text":"timestamp_update:2019-03-20T10:12:38.352Z type:TEXT content tier:3, data provider:
       Opera Institute of the Italian Vocabulary data provider:Istituto Opera del Vocabolario
        Italiano provider:CulturaItalia provider:CulturaItalia subject:Critical editions
       subject:Edizioni critiche title:Sovrana ballata piacente, title:Sovrana ballata
       piacente, title:Great nice ballad"
}
```


Figure 3.4: Example of Processed Europeana metadata for neural models

After this process, we get documents as shown in Figure 3.3. We can see that this document is now split into provided, enriched, and translated sections. However, this representation is only used for our instance of Solr-BM25, following Europeana's configuration. Solr operates on structured XML metadata, indexing various metadata fields separately. The neural models, as we configure them, require the data to be structured as free text, as they cannot take xml data as input. Unlike BM25, which retrieves documents based on discrete metadata fields, neural models work by learning dense vector representations of text. These models expect semantically meaningful passages, rather than individual metadata fields stored in separate index entries [23]. This is shown in figure 3.4

For example, for the PET dataset we merge the provided, enriched, and translated data into one cohesive textual field, to ensure that the model can capture the full context and complementary information offered by each segment. This unified representation enables the neural model to learn dense vector embeddings that encapsulate nuances from the metadata sources. We also chose to add field names to the data in a "field:data" structure. This provides explicit context that could possibly help neural models distinguish between different types of information. As opposed to simply concatenating the data, which can blur the semantic boundaries between fields—this approach preserves the inherent structure of the original metadata by clearly marking the purpose of each text segment. By labeling data elements the model can possibly better understand the semantic role of each component.

Efficiency experiment for final sample size
Given that we have around 4 million documents in our representative set, we still need to figure out if working with this amount is feasible. And so we conduct a series of experiments, given the resources we have at Europeana and at TU Delft, to determine the final sample size.

The efficiency experiment evaluates the feasibility of implementation and scalability of the models by measuring indexing time, index size, and query performance across increasing collection sizes. This helps assess how well each model scales with available computational resources and provides insights into the infrastructure required to support NIR workloads in a production setting.

The experiment is conducted on the PET document augmentation, which contains the largest amount of data in terms of both size and content. This choice ensures that the results provide an upper-bound estimate of computational requirements.

For the experiments, we measure:

- Indexing time as the number of indexed documents increases in increments of 1,000, 10,000, 100,000, 250,000, 600,000, and 1,000,000 documents.
- Index size for each of these collection sizes.
- Query time by executing 100 random queries (sampled from the click dataset), each retrieving 1,000 documents from the indexed collections.

Starting with a maximum of 1 million documents served as an appropriate upper bound for evaluating the scalability and performance of the models under experimental conditions.

Europeana RND-3:
We did the first round of experiments using Europeana's RND-3 server which is an internal system meant for testing.

Since we were using Docker to create the milvus databases it was difficult to measure the size for each collection individually because Docker's containerized environment abstracts storage usage, making it challenging to directly monitor and attribute disk space consumption to individual collections. This limitation hindered tracking of the storage overhead for each indexed collection, which is critical for assessing scalability and resource utilization in Milvus. Therefore for this set of experiments we were unable to get any data on the size of the collections.

When conducting the experiments in RND-3 we ran into another issues. The rnd-3 could not load the Jina-Colbert v2 model because one of the components of the model required a GPU with an Ampere architecture and the GPU on the server was based on the Pascal architecture (NVIDIA GeForce GTX 1080), which lacks support for features required by Ampere-based models, such as Tensor Cores optimized for mixed-precision computations. Thus we were unable to conduct the experiment for the Colbert model.

From the experiments conducted on the RND-3 we could only obtained results for the BM25, Hybrid, and Sbert models for their indexing and querying times. We could not calculate the index sizes.

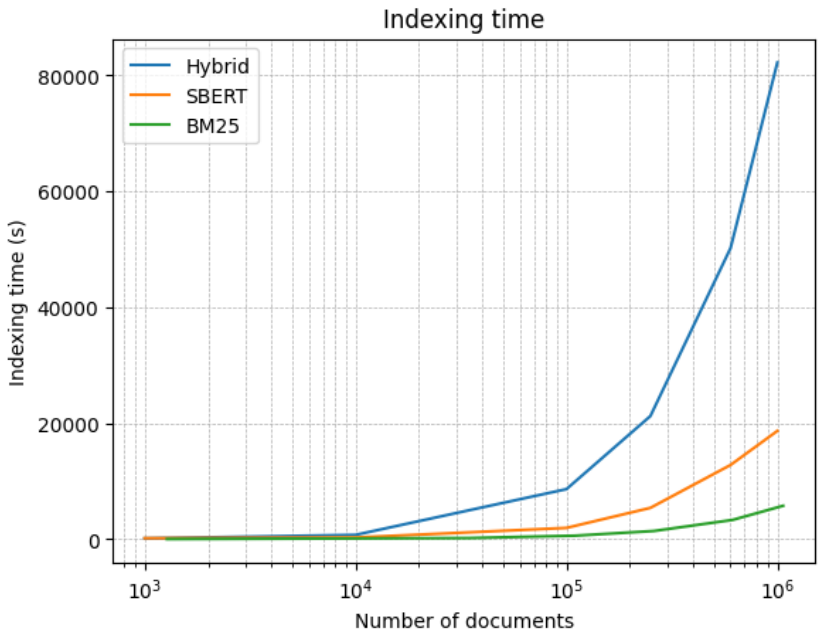


Figure 3.5: Index time vs number of documents

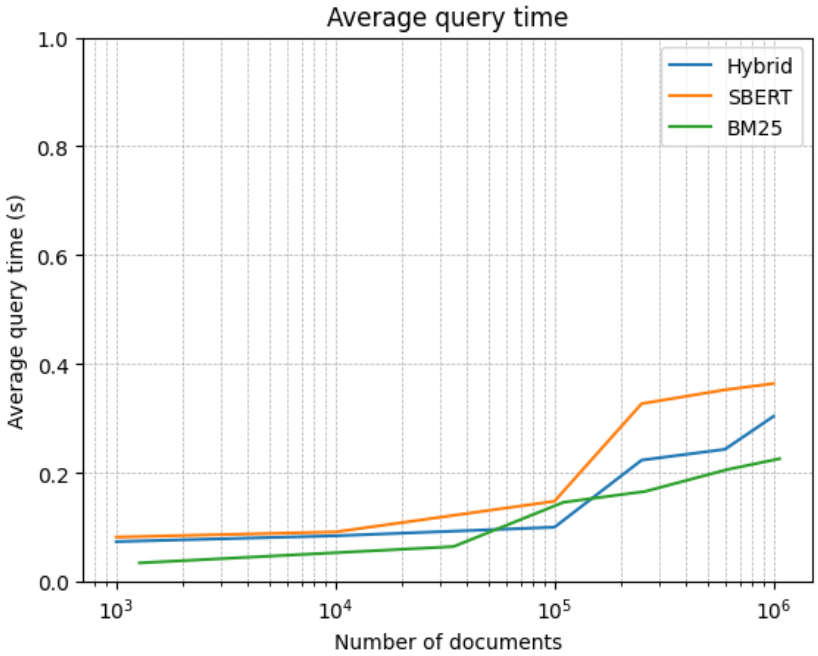


Figure 3.6: Average query time for 1000 docs vs number of documents

The Hybrid model, while demonstrating consistent and fast query performance, proved to be highly infeasible for large-scale indexing on the RND-3 infrastructure. Indexing 1 million documents took nearly an entire day, making it impractical for real-world use on this level of infrastructure. In contrast, the SBERT model, which leverages chunking, allows faster indexing, but the increased index size resulting from chunking significantly impacts query performance, causing slower retrieval times. BM25 which is the baseline system clearly works well within this infrastructure as it only takes 90 minutes to index 1000000 documents and the query times are very fast.

These findings highlight that, for scalable and efficient experimentation at Europeana, a stronger computational infrastructure is essential. The RND-3 server lacks the necessary GPU capabilities and processing power to support the Colbert model, limiting the feasibility of conducting large-scale NIR experiments within this infrastructure.

Delft AI Cluster
Since it was infeasible to run ColBERT on Europeana's infrastructure we switched to the Delft AI Cluster (DAIC) to conduct the same experiments. DAIC provides a significantly more capable infrastructure, including modern GPUs (Nvidia A40s A100s etc) and larger memory, which are essential for supporting Neural Information Retrieval (NIR) running ColBERT. By conducting the experiments on DAIC, we aimed to demonstrate the performance of NIR models on a much stronger infrastructure than RND-3.

In DAIC we were unable to run Milvus GPU via docker since its an HPC, and had to resort to Milvus Lite. This would potentially impact the rate of indexing and retrieval, but provided us the ability to measure the sizes of the indices

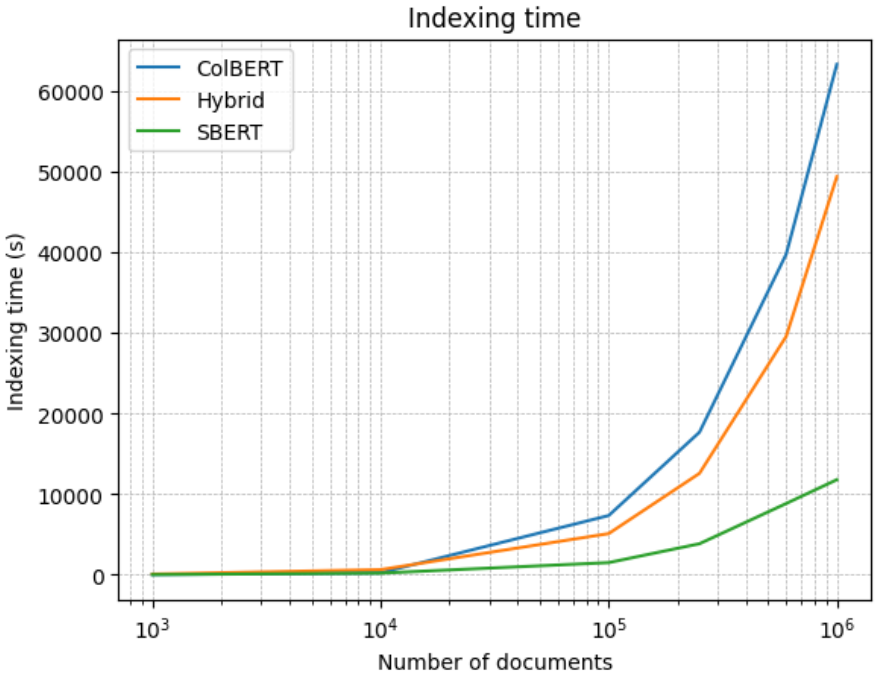


Figure 3.7: Index time vs number of documents

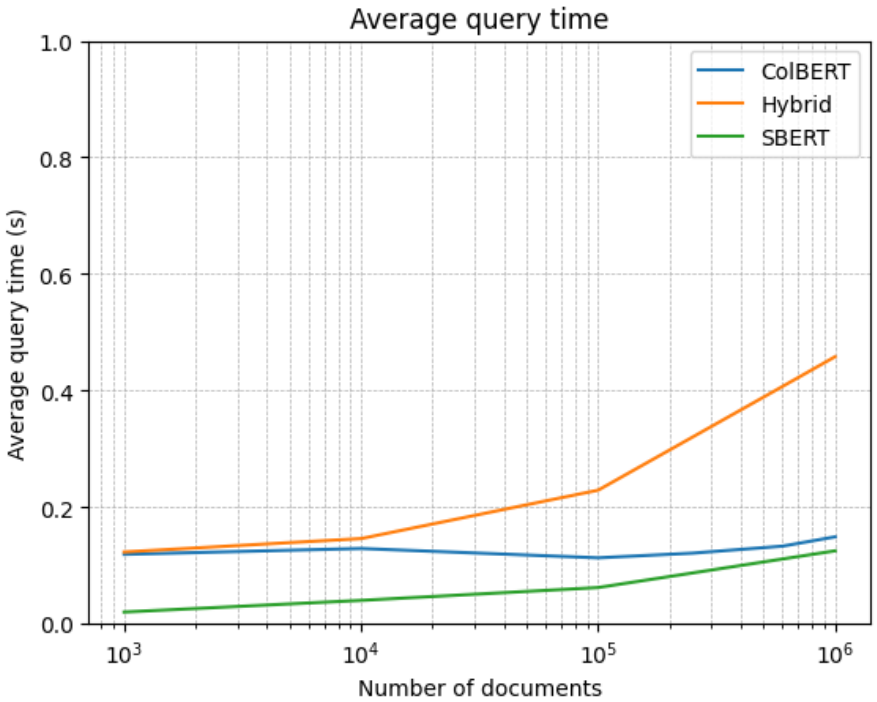


Figure 3.8: Average query time for 1000 docs vs number of documents

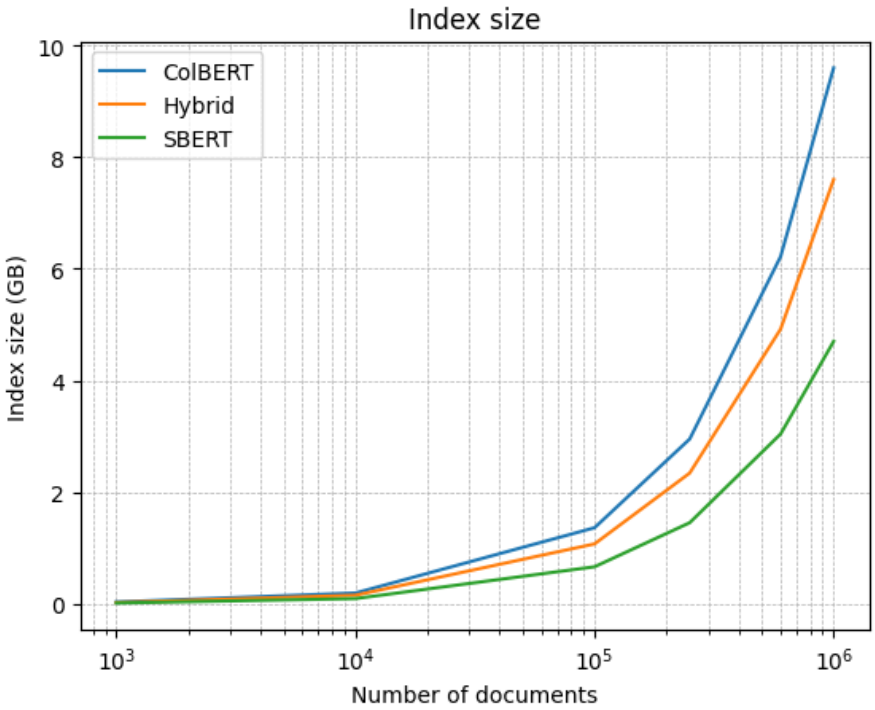


Figure 3.9: Index size vs number of documents

From the plots above we can see that DAIC queries and indexes faster than the RND-3 server even with its limitations for Milvus. Also we are able to run Jina-Colbert v2 without any issues in this environment.

In figure 3.7 we can see that the indexing times for Hybrid and SBERT models are significantly faster at 50000 seconds and 10000 seconds to index 1 million documents each. The Colbert model takes more time than the other two models, 65000 seconds for a million documents. This can be attributed to its large input size and its more fine-grained nature as it computes an embedding for each token in the input. Figure 3.8 shows us the upper limit for querying for 1000 documents and we can see that it is on par with the querying on RND-3, meaning that the querying for these models would be similar for an infrastructure between RND-3 and DAIC. Finally, figure 3.9 shows us the increase in size which goes up till 10GB for a million documents for ColBERT, 8GB for hybrid, and 5GB for a million documents for SBert. Overall, while ColBERT offers strong retrieval performance, its higher indexing times and larger index size can be attributed to its more complex representation. In contrast, SBERT and the Hybrid model demonstrate faster indexing and a smaller footprint.

Based on these results, indexing and querying 1 million documents across all models is feasible within the DAIC setup, providing a good foundation for experimentation. The manageable indexing times, query speeds, and index sizes ensure that this scale effectively balances computational efficiency with meaningful evaluation. Therefore, 1 million documents serve as an appropriate sample size for our experiments.

We obtain the 1 million documents by going over the representative set, and randomly sampling by 25%, stratifying by dataset to ensure a similar language distribution as 3.2.

3.2.2. Queries

Queries dataset

Once again, none of the queries and click data we used for the experiment were available before the investigation or were handed directly to us. Instead, with the help of the supervisor at Europeana, who provided the click data logs we were able to create a dataset for the queries.

Thus, for this investigation, we distinguish between two types of query augmentations: original queries (as issued by the user) and translated queries (translated into English). As outlined in section 3.1. This

distinction is particularly important in the context of multilingual retrieval, as it allows us to explore how translation of queries affects retrieval effectiveness; whether including the translation allows for more language diverse and relevant results. By leveraging these translations, we can assess whether translated queries improve search performance compared to their original counterparts. If translated queries lead to better retrieval outcomes, it suggests that translations serve as a useful mechanism for improving search interactions with Europeana's document collection.

The click logs was provided by Europeana and contains approximately 46000 query-document pairs. Each pair represents a user-issued query and the document that the user clicked on in response. In addition to the clicked document, the dataset includes information such as the rank at which the document appeared in the search results, the dataset to which the document belongs, and its item ID.

To complement the click logs, Europeana also provided additional query metadata. The queries metadata includes the query ID, the language of the query, and the number of search results returned for that query within the click logs. Meanwhile, the click logs itself contains the clicked document's ID along with the document language.

We used this metadata to align the query languages with the document languages which we specified in section 3.2.1, where we only included queries in that were of a language specified in our list of 20. From the click logs, after filtering for language, there are 23,000 unique queries, of which approximately 15,000 were originally issued in languages other than English and subsequently translated into English. We see the distribution in the following figure 3.10.

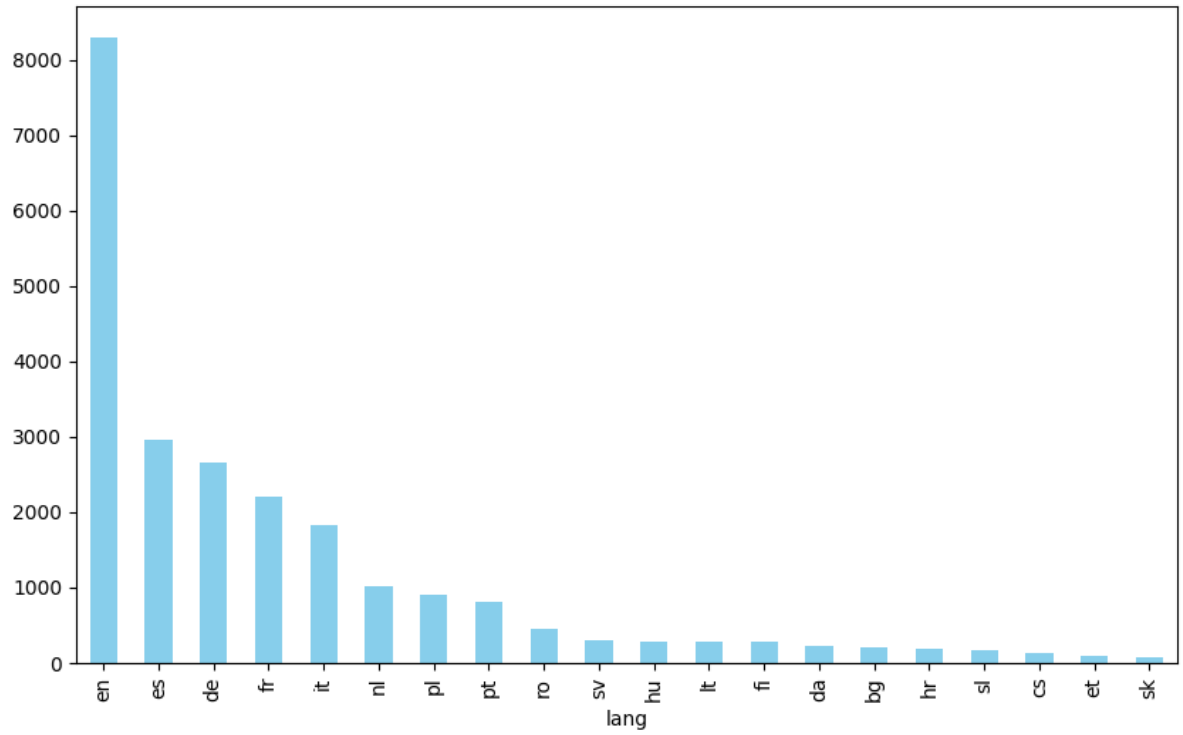


Figure 3.10: Distribtion of languages for the queries; retrieved from click logs

These translated queries were provided by Europeana, with translations generated using the Google Translate API as part of a pilot project for their Spanish portal. As a result, we were able to create two query files: an original query file and a translated query file. The original query file contained only the query ID (qid) and the query text, while the translated query file included the qid, the query text and an English translation for the non-English query. If a query was already in English, its translated field was marked as 'NaN', indicating that no translation was applied.

3.2.3. Judgements

A fundamental way of evaluating search systems is using relevance judgements, explicit labels, indicating which documents are relevant to which queries. In our investigation, we do not have any such labeled data. In an ideal scenario, Europeana would provide expert-curated relevance assessments allowing for a precise evaluation of retrieval performance.

In their absence, we adopt click logs as a proxy for relevance judgements. This method assumes that clicked documents are relevant to the query, while non-clicked documents are irrelevant. However, this assumption introduces significant uncertainties and potential biases.

There are several key limitations to consider. Clicks do not necessarily indicate true relevance. Users may click on a document for various reasons. Conversely, a highly relevant document might go unnoticed if its not been clicked. Search result rankings influence user clicks. This means that clicks are not purely a reflection of document relevance. The dataset only includes clicked documents, meaning that any document that was not clicked is implicitly treated as non-relevant. However, there could be many highly relevant documents that users simply did not encounter or click on. Furthermore, the click data aggregates interactions across multiple users and sessions.

Despite the limitations and biases associated with using click data as a proxy for relevance judgements, it remains a pragmatic choice for this investigation. Click data, though imperfect, directly reflects real-world user behavior, capturing how users interact with search results in a live system. This makes it valuable for evaluating practical retrieval performance.

In the end we create a qrel file based on the click logs. The qrels file is created by using the Clicks dataset. The data set contains all query-document pairs which represent clicks. We use this to create a file which is formatted as: <query_id> 0 <document_id> <relevance>. To ensure consistency and relevance, we filtered the click dataset to include only queries that were in one of the 20 selected languages (as outlined in Section 3.2.1). After filtering, the final qrel file contains around 45,000 query-document interactions.

It is important to acknowledge that the conclusions drawn from this study are conditioned on the assumption that clicks, despite their imperfections, contain useful signals of relevance. The broader implications of these limitations are revisited in later sections when discussing the evaluation methodology and the interpretation of results.

3.2.4. Data splits

In order to obtain the fine-tuning data we must first create a train-evaluation split from the query-document pairs from the click data, using the training queries to collect positive and negative samples, and finally form training triples with those samples.

Training-evaluation split

To construct the training triples, we first split the Clicks dataset into a training and evaluation set. The split was designed to ensure a balanced and representative distribution of multilingual data while preventing any overlap between training and evaluation queries.

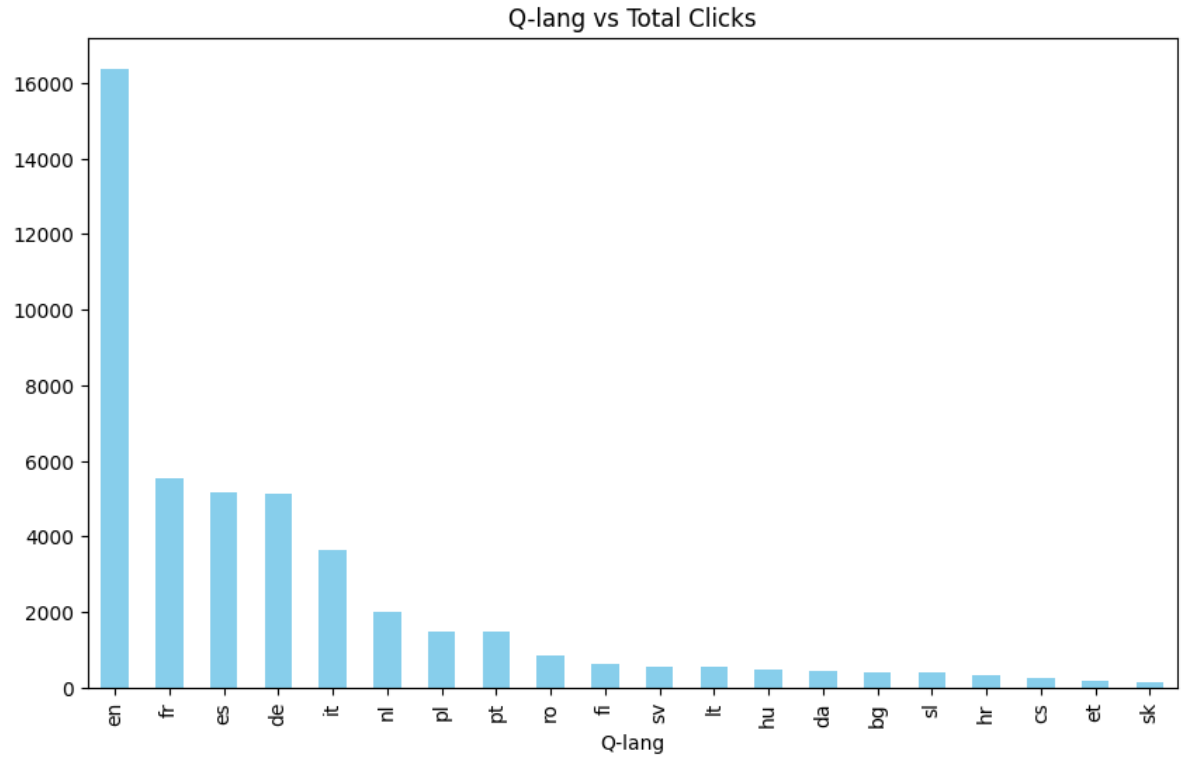


Figure 3.11: Total distribution of clicks per language

When looking at the distribution of languages per clicks we see that there is a huge bias towards english queries and thus possibly, english documents; as seen in figure 3.11. We had to be careful when forming the train and eval sets because the objective for the fine-tuning is to not only teach the model about Europeanas domain but also to tune them to Europeanas multilingual code-switched documents.

Thus we had to ensure that the training data did not have an over-representation of English documents as it might bias the model towards English and reduce its effectiveness on lower-resource languages. To mitigate this, we ensured that the training set maintained a diverse representation of language pairs while allowing the evaluation set to include as much multilingual data as possible.

This process was guided by a targets.csv file provided by the supervisors, which specified the desired number of queries and clicks for each query-document language pair (Q-lang, D-lang) in the evaluation set. The training set was then formed from the remaining queries that were not assigned to evaluation. This target file ensured that English queries-doc pairs, which were very dominant in the click dataset, would be randomly down-sampled to avoid bias in the training.

Unlike a traditional fixed 80-20 random split, the assignment of queries to the evaluation set was done incrementally, prioritizing queries that helped meet the predefined targets for each language pair. If a query contained clicks that contributed to an underrepresented language pair, it was included in the evaluation set until the target was met. In cases where a language pair had very few clicks, at least one click was always assigned to evaluation to ensure that no language combination was entirely missing for evaluation.

Since queries were assigned based on language-specific target fulfillment rather than a strict percentage, the final split did not always result in an exact 80-20 ratio. Some language pairs with more available data may have had a higher proportion allocated to training, while others with fewer available clicks may have had a larger percentage allocated to evaluation in order to meet the predefined targets. This approach ensured that low-resource languages were sufficiently included in evaluation, while high-resource languages remained well-represented in training.

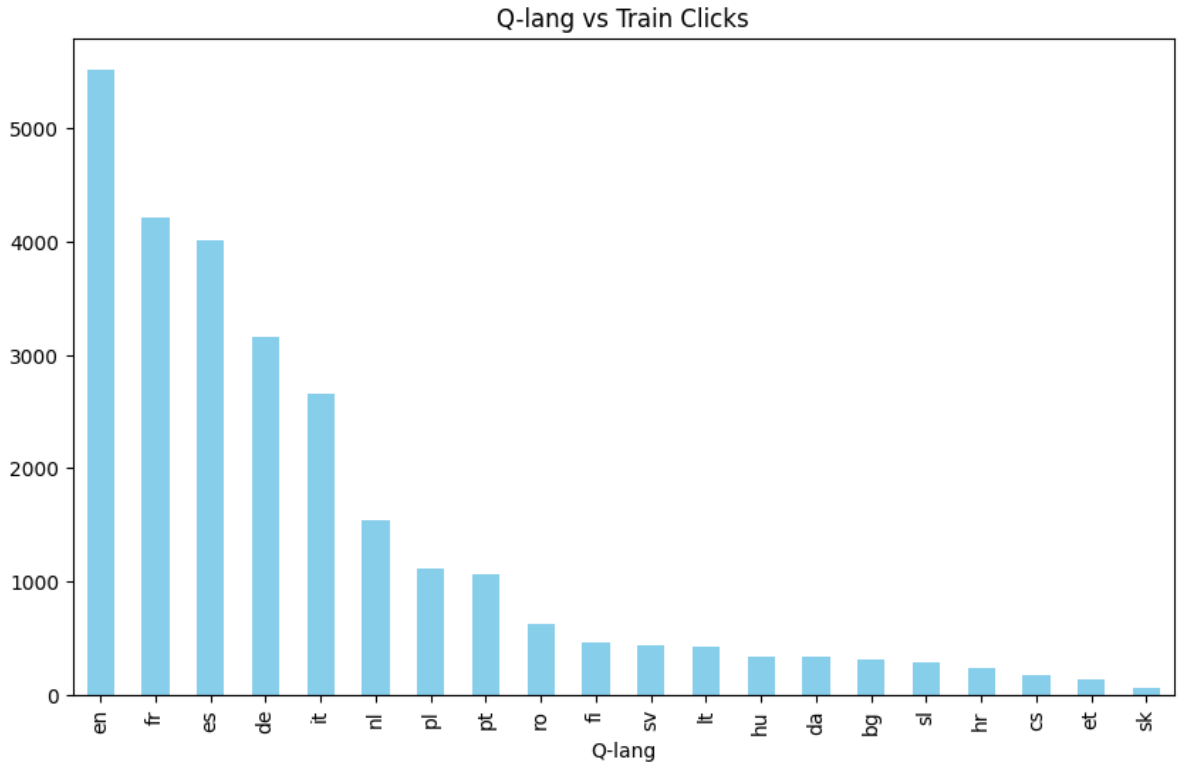


Figure 3.12: distribution of clicks per language for train data

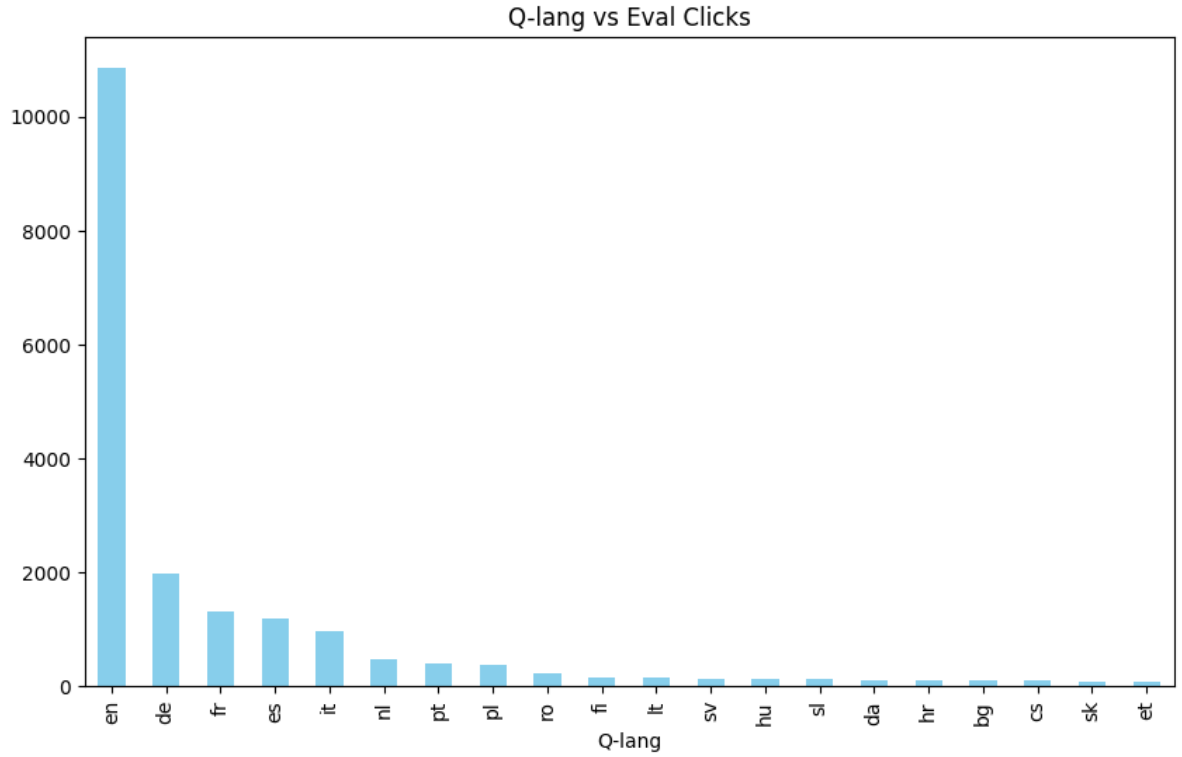


Figure 3.13: distribution of clicks per language for eval data

By following this method, we maintained linguistic diversity while adhering creating non english biased

datasets for model training and evaluation. We plot the distributions of the training and eval data per language in figure 3.12 and figure 3.13. We see that the English clicks in the traning set are not as skewed as they were in the clicks dataset. The evaluation set does have a large bias towards English; we account for this in the Evaluation.

Negative sampling and creating triples for fine-tuning

Given the data splits, the positive and negative samples were obtained using the Click data and BM25. The positives were taken from the click data where every document clicked on by the user for a specific query is taken as a positive sample.

We indexed the entire document dataset into Solr using the Europeana schema figure 3.3 for the negative samples. We only index the provided and enriched segments into the dataset as we do not want to bias the BM25 retrieval based on English with the translations. For this purpose, we deployed a different instance of Europeana's Solr-BM25. This alternative setup utilized a logical OR operator for query terms (instead of the typical AND operator used in production) ensuring a broader retrieval of potentially related documents. This approach increases the likelihood of retrieving documents loosely associated with the query while maintaining some lexical relevance.

Using this tailored BM25 instance, we executed the queries and sorted the results by ascending relevance scores. By taking the top 10% of these results (i.e., the documents with the lowest relevance scores), we effectively identified the least relevant documents retrieved by BM25 for the given query. We then select for each query a negative example from each of the 20 languages, if it existed. Finally, we form a triple by combining all positives with all negatives.

For a given query Q with N positive samples derived from the click data, and M negative samples retrieved by querying Solr, we construct $N \times M$ triples per Q.

Overall, the objective of fine-tuning was to train the model on Europeana's domain and multilinguality. Our approach is effective because positive samples were derived from user click data, ensuring alignment with real-world relevance judgements within the cultural heritage domain. This guarantees that the model learns to rank documents based on actual user preferences and interactions. For negative sampling, selecting one negative example for each target language, the model is exposed to multilingual examples, possibly compelling it to develop robust representations that can generalize across languages, especially with code-switched data. This strategy can reinforce the model to navigate Europeana code-switched multilingual datasets while learning to differentiate between positive and negative examples.

An important consideration for fine-tuning is the alignment between the data used for fine-tuning and the data used for evaluation, as emphasized in the literature. In this investigation, we ensure this alignment with respect to the document dataset augmentations: provided, enriched, and translated. Specifically, we use fine-tuning triples with the same data augmentation structure for each document augmentation indexed into a model. For instance, when evaluating a model with the PT (provided and translated) document augmentation, the fine-tuning triples for that model includes only provided and translated data. This approach ensures consistency between the fine-tuning and evaluation phases, allowing us to accurately assess each model's performance under specific augmentation scenarios.

To clarify, only the provided and enriched document augmentations are queried into the alternate Solr-BM25 instance and used during triple selection, while the documents comprising the triples can include all augmentations. This is because the P and E augmentations encompass all of the data within the documents - the translations are repetitions of some of the existing fields from the provided or enriched sections simply translated into English. Which we do because excluding the translated data during triple creation avoids introducing an English bias when querying for negatives.

3.3. Implementation

This section outlines the technical choices for the final implementation.

Vector Databases

Milvus and FAISS were chosen for storing and indexing the vectors generated by the neural models. A benefit of both tools is that they support GPU acceleration, significantly enhancing the speed of vector

indexing and querying, especially for large-scale datasets.

In this project, Milvus is prioritized for its ability to handle large-scale real-time queries and high-dimensional vectors in models like BGE-M3 and SBERT. FAISS is only used for Colbert due to its superior handling of token-level embeddings and it being pre-built into the ColBERT library. Milvus is not ideal for multi-vector embeddings because it lacks efficient support for token-level granularity.

Milvus was initially implemented using Milvus-GPU using a docker container and was later transitioned to Milvus-Lite in an Apptainer environment on the DAIC cluster. This change was necessary we could not run Milvus-GPU on the cluster. And so GPU-accelerated Milvus was unavailable.

Model specifications and parameters
Europeana's implementation of Solr-BM25 will be used with their same configuration set to create the baseline collections. As previously stated, the same data used on the other models will also be used for Solr-bm25. However, the data will not be formatted the same but instead will follow Europeana's schema compatible with Solr as shown in 1.1. This schema reflects the structured metadata format used in Europeana's existing Solr setup.

Jina-ColBERT v2 is implemented using Stanford-FutureData's ColBERT library with the Jina-AI ColBERT V2 multilingual checkpoint. FAISS-GPU is used for indexing due to its built-in integration with ColBERT and support for token-level embeddings. Data is indexed in TSV format with numeric IDs to ensure compatibility. Product quantization (PQ) is applied to optimize memory usage, with parameters set to `nbits=1` and `kmeans_niters=2` for efficient token-level retrieval. Queries are processed using the ColBERT `Searcher` class, supporting retrieval in both original and translated queries. Another major decision pertained to the max_doclen parameter, which determines the maximum number of tokens the model can process. While Jina-ColBERT v2 supports sequences up to 8192 tokens, this caused GPU memory exhaustion during indexing and retrieval. To ensure stable performance, we lowered the max_doclen to 2048 which just fit into the GPU's memory. This is a significant reduction that results in very large documents being truncated. However, given that our dataset does not contain very large documents, this adjustment may mitigate the impact on retrieval effectiveness. Nonetheless, reducing the max_doclen to 2048 was necessary to ensure the feasibility of using ColBERT within available GPU resources.

Hybrid BGE-M3 is implemented using Milvus, as it supports hybrid retrieval of sparse and dense vectors. The model produces a 1024-dimensional dense vector for semantic retrieval and a sparse lexical vector for keyword-based matching. The indexing strategy employs `IVF_PQ` with inner product (IP) for the dense vector and `SPARSE_INVERTED_INDEX` for the sparse vector. Reciprocal Rank Fusion (RRF) is used to merge retrieval results from both modalities, ensuring balanced integration of semantic and lexical relevance.

SBERT is implemented using Milvus for dense vector storage. The model generates 512-dimensional dense embeddings via `distiluse-base-multilingual-cased-v2` from SentenceTransformers. Due to its 128-token input limit, documents are chunked into smaller segments while preserving context. Retrieval is performed using inner product (IP) similarity, and chunk-level results are aggregated at the document level using the averageP method. To mitigate retrieval bias, we retrieve `k×3` results per query before aggregation and select the top 100.

To summarize, we have the following table:

Model	Indexing Method	Vector Dim.	Retrieval Strategy	Quantization	Notes
BM25	Inverted Index	N/A	Lexical BM25 retrieval	N/A	Follows Europeana's Solr schema
ColBERT	FAISS-GPU	Multivector Token-level embeddings	Inner Product (IP)	PQ (`nbits=1`, `kmeans_niters=2`)	max_doclen = 2048
Hybrid	Milvus lite	1024 (Dense) + Sparse	Inner Product (IP)	IVF_PQ and Sparse Inverted Index quantization	RRF fusion for dense + sparse retrieval
SBERT	Milvus lite	512 (Dense)	Inner Product (IP)	VF_PQ (dense)	Chunk aggregation using avgP, retrieves `k×3` before aggregation

Table 3.3: Model specifications and parameters

Fine-tuning specifications

For the fine-tuning of our retrieval models, we adopted tailored approaches for each model based on their documentation.

Jina-Colbert v2 utilized the Standford library's built-in trainer, we fine-tuned only the model component, excluding the ranker, which is managed by a separate class not involved in this process. The training data comprised of a triples.jsonl, collections.csv, and queries.csv. We ensured that alphanumeric IDs were appropriately mapped, aligning line indices with passage IDs (pids). Each triple was structured as [qid, pid+, pid-]. Given that the library is designed specifically for retrieval and triple training, we adhered to the default triple loss function (which was not customizable).

For the Hybrid model we employed distributed training via PyTorch's torch.distributed.run utility. In this process only the model is fine-tuned not the ranker, Milvus, which is not involved in this training. The fine-tuning process focused on adapting the BAAI/bge-m3 model using triples consisting of a query, a list of positive passages, and a list of negative passages. The triple data was formatted as [qid, pos_doc_list, neg_doc_list]. The library, being tailored for retrieval and triple training, employed its default loss function: m3_kd_loss.

For SBERT we leveraged the Sentence Transformers library's trainer for fine-tuning. Similar to the hybrid training, in this process only the model is fine-tuned not the ranker. Triples were formatted using a dataset dictionary from the Datasets python library, as required by the transformer library, structured as [qid, positive_doc, negative_doc]. This training utilized triplet loss, where, given a triplet of (anchor, positive, negative), it minimizes the distance between the anchor and positive while maximizing the distance between the anchor and negative. The loss is computed as: loss = max(||anchor - positive|| - ||anchor - negative|| + margin, 0).

To summarize, we have the following table:

Model	Training Framework	Training Data Format	Loss Function	Notes
Jina-ColBERT v2	Stanford ColBERT Trainer	`[qid, pid+, pid-]`	Default triplet loss (non-customizable)	Fine-tunes model only, excludes ranker
Hybrid BGE-M3	PyTorch Distributed Training	`[qid, pos_doc_list, neg_doc_list]`	`m3_kd_loss`	Fine-tunes model only, excludes Milvus (ranker)
SBERT	Sentence Transformers Trainer	`[qid, positive_doc, negative_doc]`	Triplet loss: `loss = max(||a - p|| - ||a - n|| + margin, 0)`	Fine-tunes model only, excludes ranker

Table 3.4: Fine-tuning specifications per neural model

Overall, each model has a different method of fine-tuning, but all use the same triples data.

Result collection
As outlined in Section 3.1.5, we evaluate 56 systems, where each system is defined by a model (zeroshot or finetuned) the documents indexed with that model, and the queries used to search that index.

When querying, we choose the top 100 results because according to the click data, 80% of user clicks are for documents are ranked within the top 100 results. This ensures that our evaluation focuses on the range of documents most relevant to user interactions, providing a meaningful comparison of retrieval effectiveness across the systems.

To store the top 100 results, we use the TREC run format [40]. The TREC run format is a standardized structure used for storing retrieval results, making it suitable for benchmarking information retrieval systems. The format is also compatible with libraries such as Pytrec_eval. By adopting this format, we ensure compatibility with standard evaluation frameworks and facilitate a consistent and fair comparison of our 56 systems.

Hardware
Europeana's RND-3 server is equipped with an Intel Core i7-7700 CPU featuring 8 cores. It offers 62 GB of RAM and a 31 GB swap partition. The server also includes an NVIDIA GeForce GTX 1080 GPU with 8 GB of VRAM, supporting CUDA 12.2. While these resources are sufficient for general-purpose testing and smaller-scale experiments, they may pose challenges for the implementation and evaluation of advanced NIR models, which often demand high memory, GPU capabilities, and scalability. Despite these potential limitations, conducting experiments on this infrastructure is critical to assess whether Europeana's current environment can support NIR effectively or if additional investments in computational resources are required.

Because we faced these issues on the RND-3 server, it became clear that Europeana's infrastructure was already limited for NIR. Therefore, we conducted the experiment on DAIC as well to get an idea of an infrastructure better suited for Neural Information Retrieval (NIR) tasks. DAIC provides a more robust computational environment with advanced hardware capabilities, enabling us to evaluate how these models perform when resource constraints are minimized. This comparison helps determine the extent to which infrastructure limitations impact the feasibility and efficiency of implementing NIR at Europeana. For our experiments we used nodes with the NVIDIA A40 gpu. Which each had 500gb of ram, of which we used a maximum of 100gb for the largest model, Colbert.

While DAIC was used for NIR models, Solr-BM25 was only run on RND-3. This was due to administrative restrictions, as setting up a Solr instance required admin access, which we had on RND-3 but not on DAIC.

3.4. Results and Evaluation

3.4.1. Metrics

Given the absence of absolute judgements in the dataset, we adopt a three-step evaluation process. While we assume that user clicks indicate relevance, we account for potential limitations in this assumption by conducting additional qualitative analyses. The following evaluation methodology enables us to assess system performance comprehensively by considering for each system the multilinguality, the pseudo-relevance, and rankings compared to other systems. By addressing these dimensions, we aim to understand the effectiveness of the separate system components.

Step 1: Language Distribution
The first step involves analyzing the language distribution of queries and retrieved documents. We organize results into a matrix where rows represent the query language and columns represent the document language. We compute these by using the 20 languages from section section 3.2.1 and constructing a 20x20 matrix. Each cell in this matrix represents the language combination of a query-doc pair. Using this matrix we are able to identify: pairs in the same language, pairs where the document is always in English (English column) but the query is not, and the remaining pairs (where the query and document are in different languages). The matrix is divided into three distinct regions:

- **Same-language pairs (Diagonal)** : Cases where the query and document are in the same language.
- **English retrieval (English Column)**: Cases where documents are in English but query is not English.
- **Other multilingual pairs (Remaining region)**: All other (different) combinations of query and document languages.

This classification helps us assess the degree of monolinguality (via the diagonal region) and the degree of multilingual retrieval (via the English and remaining regions).

Metrics for this step include:

- **Counts**: Number of results in a region.
- **Percentage**: Percentage of results in a region.
- **Entropy**: Diversity within each region to evaluate language distribution effectiveness.

Key expectations include:

- **Counts and percentages**: We expect higher counts in the "same language" region for baseline systems (BM25) due to their reliance on lexical matching. In contrast, neural models are expected to distribute counts more evenly across the "same language," "English," and "different language" regions, particularly when query and document augmentations are applied. We anticipate the multilingual capabilities of the neural models to enhance performance both with and without the augmentations.
- **Entropy**: Entropy measures the diversity in language distribution, and we anticipate:
  - H_{same}: We expect H_{same} to increase slightly with augmentations, as they improve the balance of retrieval across languages.
  - H_{en}: We expect H_{en} to increase for systems using document translation, as translations enhance diversity in English-language retrieval by increasing the number of translated documents, allowing for the systems to retrieve english documents even from non-english queries.
  - H_{diff}: We expect H_{diff} to rise with document enrichment and translation, reflecting improved inclusion of documents in diverse languages.

Step 2: Rank comparison
The second step focuses on examining the ranking similarity between systems for every query. This comparison evaluates the impact of individual augmentations—such as query augmentation, document enrichment, fine-tuning, and model selection—on the rankings produced by the systems. The primary metric for this step is: **Rank-Biased Overlap (RBO)**

RBO is a similarity measure designed to compare two ranked lists, accounting for elements in common and emphasizing top-ranked items [11]. It is used for analyzing ranking differences in information retrieval tasks, such as comparing rankings generated by different retrieval systems for the same queries.

For this analysis, we use an extended version of RBO [11], which handles ties effectively by treating tied rankings as equal rather than uncertain. This is achieved by using the $w - variant$ RBO, ensuring that documents with the same score are treated fairly in the similarity calculation. Unlike traditional ranking metrics, RBO accounts for incomplete or indefinite rankings and allows for partial matches, making it well-suited for information retrieval tasks.

The calculation of RBO incorporates a persistence parameter (p), which determines how much weight is given to higher-ranked items. A higher p value (commonly set to 0.95) places greater emphasis on the top of the rankings, aligning with the idea that the highest-ranked documents are often the most important. RBO is also robust to differences in ranking depths and can handle ties effectively. This is useful for us since comparisons for query results between Solr-BM25 and neural models are often between ranks of different lengths.

The RBO is calculated using the RBO library provided in[11]. We use the libraries $extract\_ranking$ method which creates a ranking of results sorting by the score in descending order and items with the

same score are grouped in a tie. We then use the rbo function which computes the RBO between two rank lists. We use the w parameter as in our case a tie represents equality of ranks and not uncertainty.

Key expectations include:

- **Query Augmentations**: RBO may reveal ranking shifts, especially for non-English queries, as translations retrieve additional relevant documents.
- **Document Augmentations**: We expect RBO to show increased retrieval diversity. Enrichments may boost English document rankings, while translations could make English results more dominant.
- **Fine-Tuning**: Fine-tuned models might show higher ranking consistency, but RBO could highlight divergence from BM25, reflecting a stronger semantic focus.
- **Model Comparisons**: RBO is likely to indicate major differences between BM25 and neural models, with BM25 rankings closely following click data, while neural models retrieve a broader, more diverse set of relevant documents.

Step 3: Click-Based Retrieval Metrics
In the third step, we evaluate retrieval performance using click data. Here we make a big decision in taking clicks as pseudo-relevance judgements. We do this because it provides a user-centric perspective on retrieval performance. Clicks indicate user interest and engagement with search results, making them a valuable implicit signal of relevance.

The evaluation is conducted over the ranked list of retrieved documents for each query, using the following metrics:

- **Average Precision (AP)**: Measures the precision values at each rank where a relevant document is retrieved for a single query. It reflects how well a system ranks relevant documents for that specific query.
- **Reciprocal Rank (RR)**: Evaluates the ranking of the first relevant (clicked) document in the retrieved list. It is calculated as the reciprocal of the rank of the first clicked document:
- **Recall**: Measures the proportion of relevant (clicked) documents retrieved out of the total number of relevant documents for a query. This metric captures how well the retrieval system identifies all relevant documents, regardless of their ranking.

Click-based metrics provide insights into ranking quality and retrieval effectiveness, assuming clicks as relevance judgements. The AP, RR, and Recall are calculated using the pytrec_eval library. This library makes use of trec run files and qrel files to calculate IR metrics as outlined in section 3.3

In this situation, all the <relevance> fields will be set to 1, which is not an issue since the metrics being calculated (AP, RR, and Recall) only require binary relevance judgements. These metrics do not differentiate between varying degrees of relevance, as their focus is on whether a document is relevant or not based on user clicks.

Using this binary relevance assumption aligns with the purpose of these metrics:

- **Average Precision (AP)**: Evaluates how well the system ranks clicked documents by averaging precision scores at ranks where clicked documents appear. A binary relevance value is sufficient to determine whether a document contributes to precision.
- **Reciprocal Rank (RR)**: Focuses on the rank of the first clicked document. Since only the presence or absence of a click matters, binary relevance is adequate.
- **Recall**: Measures the proportion of clicked documents retrieved. Binary relevance ensures that all clicks are considered equally for calculating the percentage of relevant documents retrieved.

The use of binary relevance simplifies the evaluation process and is consistent with the assumptions underlying the metrics, making it appropriate for assessing system performance based on click data.

Key expectations include:

- **Average Precision (AP)**: We expect high AP to indicate that the system places clicked documents closer to the top of the ranking. Neural models might show lower AP compared to BM25 due to the latter's alignment with the click data, but neural models may still surface relevant but previously unseen documents.
- **Reciprocal Rank (RR)**: We expect high RR to reflect that the system ranks at least one clicked document very high, demonstrating good performance for queries where immediate relevance is crucial.
- **Recall**: We expect high recall to show that the system retrieves all clicked documents, regardless of their rank. We expect BM25 to have higher recall due to the bias of the data towards the ranks.

BM25's bias stems from the click data being generated using a similar Solr-BM25 system, meaning users interacted primarily with documents ranked highly by BM25. This bias implies that BM25 is likely to outperform neural models on metrics like AP and RR because it is inherently optimized for the same retrieval patterns that produced the click data. However, this does not necessarily mean BM25 provides better overall retrieval quality.

For neural models, lower AP or RR may reflect their ability to retrieve documents that BM25 did not surface but which could be relevant based on semantic similarity. As a result, while BM25's alignment with the click data gives it an advantage in precision-focused metrics, neural models may excel in recall, demonstrating their capability to relevant content but possibly at lower ranks. This outlined why we considering multiple dimensions, not only rank metrics, to evaluate the retrieval systems.

3.4.2. Collecting the results

For each system, we run all 9,100 queries from the evaluation set. In the case of the no query augmentations (using the original queries) we simply run the singular queries on the models. In the case of query augmentations, when we have to search with the original and translated queries, we approximate Europeanas methodology of doing an 'OR' search. For BM25 this entails formatting the query as [{query_original} OR {query_translated}], which is how searches are conducted with logical operators in Solr. For neural models a logical OR search is not possible since the vector databases being used do not support search with logical operators, therefore, we launch two distinct searches and combine the results using RRF. Although the Hybrid model already uses RRF (as outlined in section 3.1.3) we apply it again between the original query and the translated query results. By applying RRF, we approximate the behavior of an "OR" search for neural models, allowing us to merge results from the original and translated query searches while balancing their contributions based on rank.

For each query, we stored the top 100 results in this format. This means each system produced up to 910,000 results, representing the theoretical upper limit. BM25 usually returns far fewer than 100 results or fail to retrieve any documents in extreme cases. Which is something we see happen often in our BM25 results. In the end we had 910000 results for all of the neural systems, while for the Solr-BM25 systems we had between 150000 and 230000 results.

Per-query handling

For many metrics, we calculate an average over all queries. However, since there is a disproportionately higher number of English queries compared to non-English queries as seen in figure 3.13, we stratify this averaging process by separating English and non-English queries. This involves calculating the metrics separately for English and non-English queries and then providing two distinct averages: _EN (English) and _NEN (Non-English). The Non-english average is further stratified per language.

This stratification allows us to better understand how the retrieval systems perform across different linguistic contexts. By distinguishing between English and non-English queries, we can identify potential biases in the systems, such as whether they are disproportionately optimized for English content. Additionally, it helps ensure that the evaluation metrics reflect the multilingual challenges of the dataset rather than being dominated by the majority language. This stratified analysis provides more granular insights into the systems' performance across languages, aiding in assessing their effectiveness in a multilingual information retrieval setting.

We apply this process this for the following metrics: AP, RR, Recall, and RBO.

4

Results

4.1. Explanation of results

We analyze retrieval effectiveness using these metrics using two tables:

1. **Absolute Metrics**: Language distribution and click-based metrics, providing insights into overall system performance to determine the best-performing system; done per system
2. **Comparative Metrics**: Comparing combinations of results from the Absolute table for each system with respect to a specific type of model, augmentation, and finetuning. For example, we would compare Solr-bm25 without query augmentation with Solr-bm25 with query augmentation (keeping the fine-tuning and document augmentation identical), in order to gauge the impact of query augmentation. We also include RBO in this table since it is comparative by default.

We perform comparisons to evaluate the effect of query and document augmentations, fine-tuning, and retrieval approaches:

- **Mode selection**: Analysis of the performance differences across BM25, ColBERT, Hybrid, and SBERT retrieval systems.
- **Fine-Tuning**: Comparison of systems with and without fine-tuning, evaluating the impact of domain adaptation.
- **Query Augmentation**: Comparison of original queries (Q) versus translated queries (Q+O), evaluating the impact of query translation on retrieval.
- **Document Enrichment**: Comparison of P (provided) or P+T (provided + translated) against P+E (provided + enriched) or P+E+T (provided + enriched + translated), assessing the value of enrichment.
- **Document Translation**: Comparison of P (provided) or P+E (provided + enriched) against P+T (provided + translated) or P+E+T (provided + enriched + translated), evaluating the contribution of translation.

For each comparison we perform the three-step evaluation to ensure a robust analysis of system performance:

- **Language Distribution**: Reveals system behavior across multilingual contexts.
- **Ranking Similarity**: Highlights system consistency and alignment across configurations.
- **Click-Based Metrics**: Provides insights into user-relevant retrieval effectiveness.

By combining absolute and comparative metrics, this process provides a holistic evaluation of retrieval systems, considering both domain-specific and multilingual retrieval challenges, while acknowledging the challenges of not having concrete relevance judgements.

4.2. Initial quantitative results

This section provides a sample from our quantitative analysis to illustrate the structure and nature of the raw results. The complete results can be found in the appendix.

4.2.1. Absolute results

System	N_{same}	N_{EN}	N_{diff}	pct_{same}	pct_{EN}	pct_{diff}	H_{same}	H_{EN}	H_{diff}
B-O-P	70131.00	68916.00	14067.00	45.80	45.01	9.19	2.46	0.67	6.16
B-O-PE	70403.00	74094.00	17041.00	43.58	45.87	10.55	2.45	0.70	6.29
B-O-PT	60528.00	108082.00	14781.00	33.00	58.94	8.06	2.72	0.84	6.22
B-O-PET	61291.00	110807.00	17707.00	32.29	58.38	9.33	2.70	0.88	6.31
B-OT-P	67394.00	86443.00	27219.00	37.22	47.74	15.03	2.39	2.33	6.71
B-OT-PE	68164.00	91530.00	29078.00	36.11	48.49	15.40	2.39	2.33	6.72
B-OT-PT	58989.00	118741.00	49604.00	25.95	52.23	21.82	2.66	2.23	6.78
B-OT-PET	60310.00	121608.00	50362.00	25.96	52.35	21.68	2.65	2.23	6.78
CZ-O-P	230634.00	463284.00	217282.00	25.31	50.84	23.85	2.67	1.79	6.86
CZ-O-PE	235488.00	461655.00	214057.00	25.84	50.66	23.49	2.70	1.70	6.84
CZ-O-PT	223877.00	468196.00	219127.00	24.57	51.38	24.05	2.72	1.79	6.86
CZ-O-PET	232548.00	465519.00	213133.00	25.52	51.09	23.39	2.74	1.70	6.84
CZ-OT-P	222067.00	467591.00	221542.00	24.37	51.32	24.31	2.63	1.91	6.87
CZ-OT-PE	225559.00	467003.00	218638.00	24.75	51.25	23.99	2.65	1.87	6.85
CZ-OT-PT	215802.00	470892.00	224506.00	23.68	51.68	24.64	2.68	1.89	6.87
CZ-OT-PET	223111.00	469643.00	218446.00	24.49	51.54	23.97	2.70	1.85	6.84

Table 4.1: Absolute results for all BM25 and Colbert systems: language distribution

System	AP_{EN}	AP_{NEN}	R_{EN}	$R_N EN$	RR_{EN}	RR_{NEN}
B-O-P	0.62	0.75	0.76	0.89	0.68	0.79
B-O-PE	0.63	0.76	0.77	0.90	0.69	0.80
B-O-PT	0.72	0.76	0.90	0.90	0.77	0.80
B-O-PET	0.73	0.78	0.91	0.91	0.78	0.81
B-OT-P	0.62	0.69	0.76	0.88	0.68	0.73
B-OT-PE	0.63	0.69	0.77	0.89	0.69	0.73
B-OT-PT	0.72	0.69	0.90	0.89	0.77	0.73
B-OT-PET	0.73	0.70	0.91	0.90	0.78	0.74
CZ-O-P	0.22	0.32	0.52	0.62	0.25	0.36
CZ-O-PE	0.23	0.34	0.54	0.64	0.27	0.38
CZ-O-PT	0.21	0.31	0.54	0.62	0.25	0.35
CZ-O-PET	0.23	0.34	0.56	0.65	0.27	0.38
CZ-OT-P	0.22	0.29	0.52	0.61	0.25	0.32
CZ-OT-PE	0.23	0.30	0.54	0.63	0.27	0.34
CZ-OT-PT	0.21	0.28	0.54	0.62	0.25	0.32
CZ-OT-PET	0.23	0.31	0.56	0.64	0.27	0.34

Table 4.2: Absolute results for BM25 and Colbert systems: performance metrics

4.2.2. Comparative results

Model comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	CZ-O-P	-0.41	-0.43	-0.24	-0.27	-0.43	-0.43	0.24	0.34
B-O-PE	CZ-O-PE	-0.40	-0.42	-0.23	-0.26	-0.42	-0.41	0.25	0.35
B-O-PT	CZ-O-PT	-0.51	-0.45	-0.36	-0.28	-0.52	-0.45	0.23	0.33
B-O-PET	CZ-O-PET	-0.50	-0.43	-0.35	-0.26	-0.51	-0.43	0.25	0.35
B-OT-P	CZ-OT-P	-0.41	-0.40	-0.24	-0.27	-0.43	-0.41	0.24	0.30
B-OT-PE	CZ-OT-PE	-0.40	-0.39	-0.23	-0.26	-0.42	-0.39	0.25	0.32
B-OT-PT	CZ-OT-PT	-0.51	-0.41	-0.36	-0.27	-0.52	-0.42	0.23	0.29
B-OT-PET	CZ-OT-PET	-0.50	-0.40	-0.35	-0.26	-0.51	-0.40	0.25	0.31

Table 4.3: Difference in performance metrics for BM25 and Colbert

Query augmentation comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	B-OT-P	0.00	-0.06	0.00	-0.01	0.00	-0.07	1.00	0.87
B-O-PE	B-OT-PE	0.00	-0.06	0.00	-0.01	0.00	-0.07	1.00	0.87
B-O-PT	B-OT-PT	0.00	-0.07	0.00	-0.01	0.00	-0.07	1.00	0.84
B-O-PET	B-OT-PET	0.00	-0.07	0.00	-0.01	0.00	-0.07	1.00	0.85
CZ-O-P	CZ-OT-P	0.00	-0.04	0.00	-0.01	0.00	-0.04	1.00	0.70
CZ-O-PE	CZ-OT-PE	0.00	-0.04	0.00	-0.01	0.00	-0.05	1.00	0.71
CZ-O-PT	CZ-OT-PT	0.00	-0.03	0.00	-0.00	0.00	-0.04	1.00	0.70
CZ-O-PET	CZ-OT-PET	0.00	-0.04	0.00	-0.01	0.00	-0.04	1.00	0.71

Table 4.4: Difference in performance metrics for original and translated queries

Document enrichment comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	B-O-PE	0.00	0.01	0.01	0.01	0.00	0.00	0.95	0.97
B-O-PT	B-O-PET	0.01	0.01	0.01	0.01	0.01	0.01	0.96	0.97
B-OT-P	B-OT-PE	0.00	0.00	0.01	0.01	0.00	0.00	0.95	0.96
B-OT-PT	B-OT-PET	0.01	0.01	0.01	0.01	0.01	0.01	0.96	0.96
CZ-O-P	CZ-O-PE	0.01	0.02	0.02	0.01	0.01	0.02	0.32	0.37
CZ-O-PT	CZ-O-PET	0.02	0.03	0.02	0.03	0.02	0.03	0.31	0.36
CZ-OT-P	CZ-OT-PE	0.01	0.02	0.02	0.01	0.01	0.02	0.32	0.35
CZ-OT-PT	CZ-OT-PET	0.02	0.03	0.02	0.03	0.02	0.02	0.31	0.34

Table 4.5: DPerformance metrics across different models and augmentation strategies

Document translation comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	B-O-PT	0.10	0.01	0.13	0.01	0.08	0.01	0.82	0.96
B-O-PE	B-O-PET	0.11	0.02	0.13	0.01	0.09	0.02	0.83	0.97
B-OT-P	B-OT-PT	0.10	0.00	0.13	0.01	0.08	0.01	0.82	0.87
B-OT-PE	B-OT-PET	0.11	0.01	0.13	0.01	0.09	0.01	0.83	0.87
CZ-O-P	CZ-O-PT	-0.00	-0.01	0.02	-0.00	-0.00	-0.01	0.32	0.37
CZ-O-PE	CZ-O-PET	0.00	-0.00	0.02	0.01	0.00	-0.00	0.36	0.41
CZ-OT-P	CZ-OT-PT	-0.00	-0.01	0.02	0.00	-0.00	-0.00	0.32	0.35
CZ-OT-PE	CZ-OT-PET	0.00	0.00	0.02	0.01	0.00	0.00	0.36	0.39

Table 4.6: Performance metrics across different models and augmentation strategies

4.3. Results Analysis

The following outlines the initial results of the experiments. Here, we have analysed the impact of each component on the system to the distribution of the languages, the rankings, and the click metrics.

For the language distribution, we will examine the percentage of documents retrieved in the same language as the query (pct_same), in English (pct_EN), and in a different language (pct_diff) than the query. This will help us understand the impact of different models on multilingual retrieval.

For ranking behavior, we use the Rank-Biased Overlap (RBO) metric to measure the similarity between ranked retrieval results. Specifically, we compare rankings between different system variations, such as with and without query, fine-tuning, or document augmentations, as well as between BM25 and neural models. A high RBO score indicates that the rankings remain relatively unchanged, suggesting that the component in question has little effect on retrieval order. Conversely, a low RBO score implies that the component significantly alters rankings, demonstrating a strong impact on retrieval behavior.

For click-based evaluation, we analyze differences in Average Precision (AP), Recall, and Reciprocal Rank (RR) when system components are modified. This includes comparisons between BM25 and neural models, as well as between different augmentation strategies. A negative difference in these metrics suggests that the change has resulted in a decline in performance, while a positive difference indicates an improvement. These results provide insight into how each component contributes to retrieval effectiveness and whether neural models successfully enhance ranking quality over BM25.

It is important to reiterate that with the click metrics, we expect to see a large bias for the BM25 model as outlined in Chapter 3.4.1. Furthermore, we do not have complete judgements since we base our judgement of relevance for a document on the user's click data and not on annotations.

4.3.1. Model choice

First, we look into the impact of the choice of model on the system's performance, keeping all of the other system components the same but only changing the model type.

Language distribution

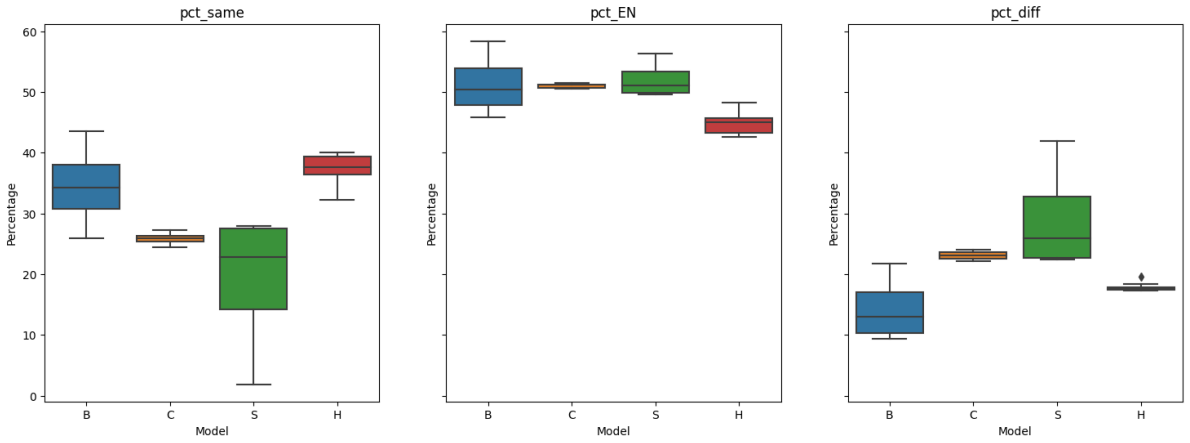
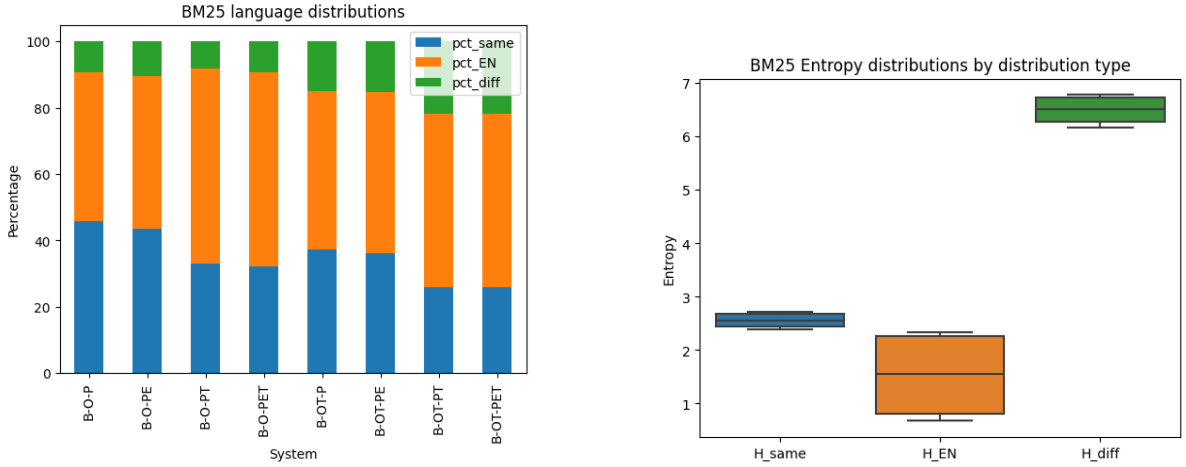


Figure 4.1: Language distribution for all models

Figure 4.1 shows us the differences in the percentage of retrieved documents. For pct_same, we can see that the BM25 and Hybrid models have the highest percentages between 30% and 40% on average. This can be attributed to the lexical matching that both of these models rely on; completely for BM25 and only partially for Hybrid. Colbert and Sbert have on average, similar percentages (Sbert does have significant outliers) at around 25% count for documents retrieved in the same language as the query. For pct_EN, BM25, ColBERT, and SBERT retrieve the highest percentage of English documents, both around 50-60%. ColBERT retrieves slightly fewer English documents, with significantly less variation, while Hybrid retrieves the lowest percentage of English documents (45%), indicating a stronger multilingual retrieval performance. For pct_diff, SBERT has the highest variability, retrieving a broad range of documents in

different languages (20-50%), while BM25 and Hybrid retrieve the fewest different-language documents (10-20%). ColBERT has a very stable distribution of around 25%, suggesting a balanced retrieval strategy. The high pct_EN across models is likely due to the overrepresentation of English documents and queries in the evaluation set, which increases the likelihood of retrieving English content.



(a) Language distribution for BM25 over all augmentations

(b) BM25 entropy value per distribution metric

Figure 4.2: Language distribution for BM25

Figure 4.2 illustrates the language distribution of BM25 across all query and document augmentations, highlighting their significant impact on retrieval patterns. Initially, BM25 retrieves 45% same-language documents, 45% English documents, and 10% different-language documents. As augmentations increase—through query translations, document enrichments, and document translations—the proportion of different-language documents (pct_diff) rises to 50%, while English documents (pct_EN) increase to 20%, reducing the same-language count to 25%. This indicates that BM25 relies heavily on augmentations to improve multilingual retrieval, as augmentations lower monolingual search by nearly 20%. Additionally, entropy analysis reveals that BM25's same-language and English document retrieval is concentrated around a few query types (likely English queries) as it is quite low 2.5, while different-language retrieval is more widely distributed across queries, 7. This suggests that augmentations help BM25 expand multilingual retrieval but may still be query-dependent in how effectively it retrieves diverse-language documents.

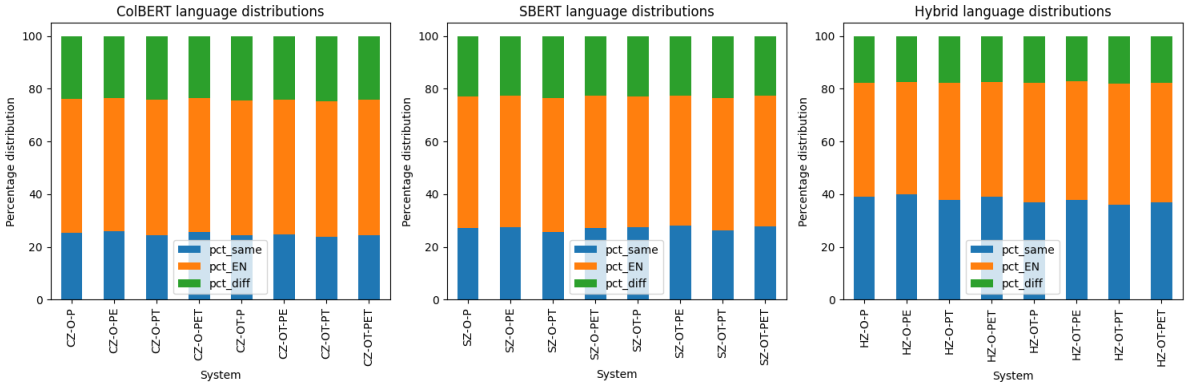


Figure 4.3: Language distribution for pre-trained Neural models

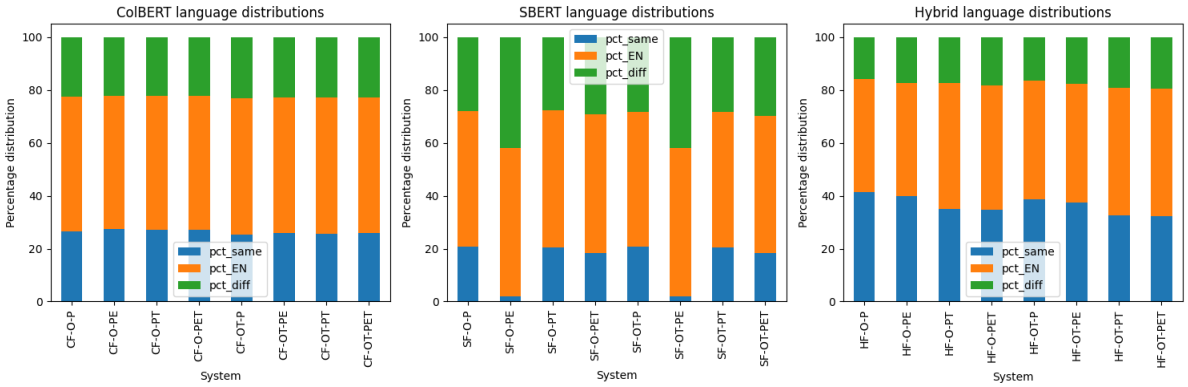


Figure 4.4: Language distribution for finetuned Neural models

Figures 4.3 and 4.4 show the distribution of the pre-trained and fine-tuned Neural models over all augmentations, respectively. Unlike BM25, which exhibits a shifting language distribution with increasing augmentations, Figure 4.3 shows us that the NIR models maintain a relatively stable proportion of same-language, English, and different-language documents as augmentations are introduced. Colbert and Sbert have similar distributions of 25% for same-language, 50% for english-language, and 25% for different-language documents. While Hybrid has a higher same-language at around 40%, english-language of 40%, and different-language documents of 20%. The higher hybrid same-count can be attributed to its lexical matching.

Further analysis of the fine-tuned NIR models in Figure 4.4 reveals varying impacts of fine-tuning on language distribution. ColBERT remains largely unchanged, maintaining a similar distribution to its pretrained version, with no noticeable shifts in language distribution after fine-tuning. Hybrid exhibits slight changes, particularly when translations are introduced, leading to a 5% decrease in same-language document retrieval and a 3% increase in English-language retrieval. This suggests that fine-tuning on translated data helps Hybrid better leverage translations for improved English retrieval. SBERT also remains stable post-fine-tuning, except in two outlying cases when document enrichments are present. In these cases, pct_diff increases significantly from 28% to 40%, while pct_EN drops sharply from 20% to 1%, indicating a major shift in distribution only when enriched documents are included.

Comparative analysis

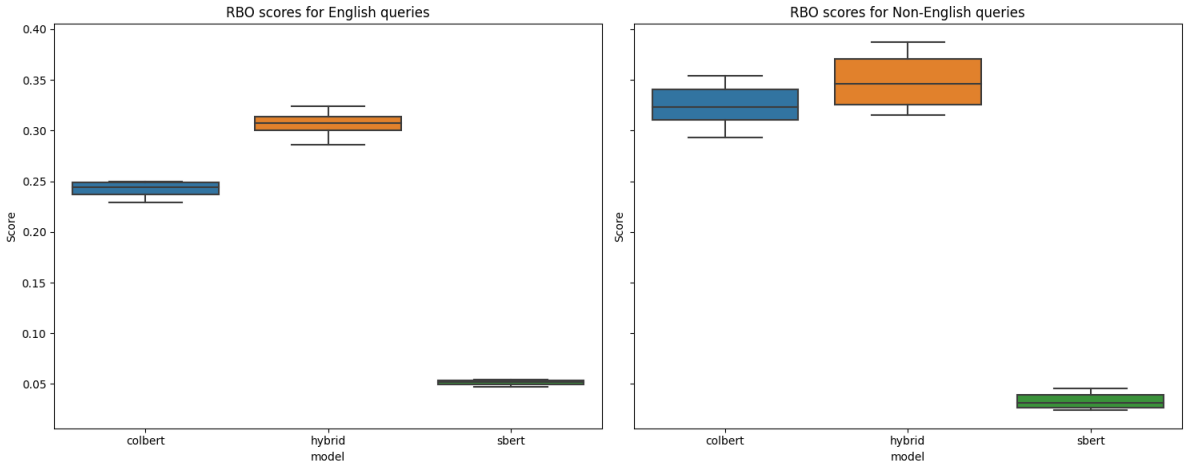


Figure 4.5: average RBO scores with respect to model comparison: BM25 vs Neural

Figure 4.5 presents the RBO scores comparing the ranking similarity between BM25 and various neural models while keeping all other system specifications constant. A high RBO score suggests that the

rankings produced by BM25 and the neural model are similar, implying that switching models would yield comparable rankings. Conversely, a low RBO score indicates a significant divergence in rankings, meaning that changing the retrieval model alone substantially alters the results. The figure reveals that, on average, RBO scores are low across all neural model comparisons with BM25, highlighting that model selection strongly influences ranking behavior and retrieval outcomes. For Colbert and Hybrid the scores are low at around 0.25-0.3 but the Sbert scores are much lower, close to 0, indicating that the ranking between sbert and bm25 are completely different.

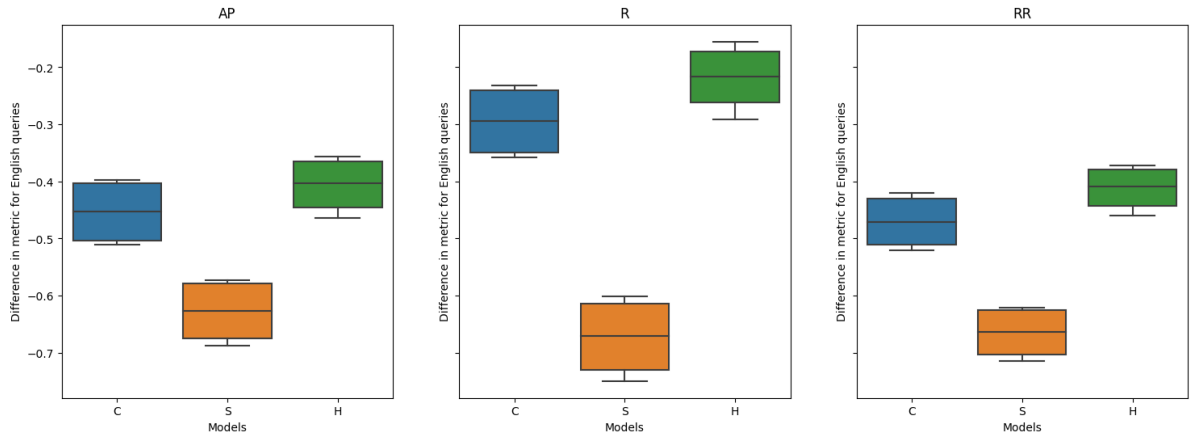


Figure 4.6: Difference in click metrics for English queries with respect to the Model augmentation from BM25 to Neural. A negative difference indicates a decline in performance after switching from BM25 to a neural model, while a positive difference suggests an improvement.

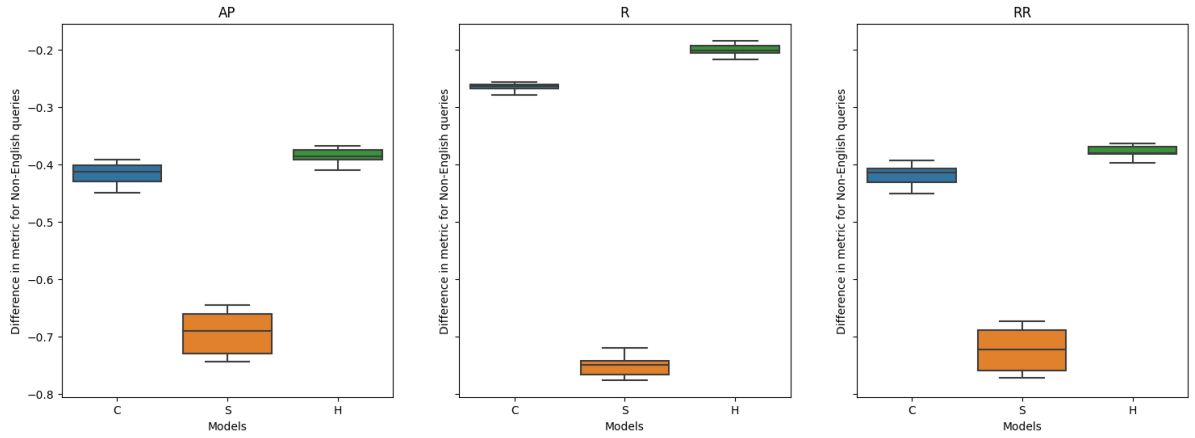


Figure 4.7: Difference in click metrics for Non-English queries with respect to the Model augmentation from BM25 to Neural. A negative difference indicates a decline in performance after switching from BM25 to a neural model, while a positive difference suggests an improvement.

To assess whether the significant ranking changes introduced by different models are beneficial, we analyze the click-based metrics outlined in figure 4.6 and figure 4.7. These metrics include Average Precision (AP), recall, and reciprocal rank, with the plots illustrating the average difference between the neural models and BM25. A negative difference indicates a decline in performance after switching from BM25 to a neural model, while a positive difference suggests an improvement. This analysis helps determine whether the observed ranking shifts contribute to better retrieval effectiveness.

The plots show, on average, a reduction in score when changing from the BM25 to the neural model. Between Bm25 and Colbert/Hybrid the reduction in metric scores are not as large as with Bm25 and Sbert. For Colbert and Hybrid, the AP and RR are reduced by 0.45 but the recall is reduced by 0.2/0.3. This indicates that while ColBERT and Hybrid models exhibit a drop in AP and RR compared to BM25,

their recall remains relatively higher, suggesting that these models still retrieve a similar number of relevant documents compared to BM25 but rank them differently than BM25. This could be because the neural models are able to retrieve a lot more semantically relevant documents which, in our biased evaluation system, are not judged properly.

The difference in performance between SBERT and BM25 is very big at 0.6/0.7, suggesting that SBERT behaves very differently for this dataset and task. Its distinct ranking patterns and alignment with user interaction metrics indicate that SBERT approaches retrieval in a way that contrasts with the other models. We cannot conclude that it is bad nor ineffective since we do not have complete judgements and the values we are using as judgements now are heavily biased towards BM25.

4.3.2. Fine-tuning impact

Now looking into the impact of fine-tuning the neural models on the system's performance.

Language distribution

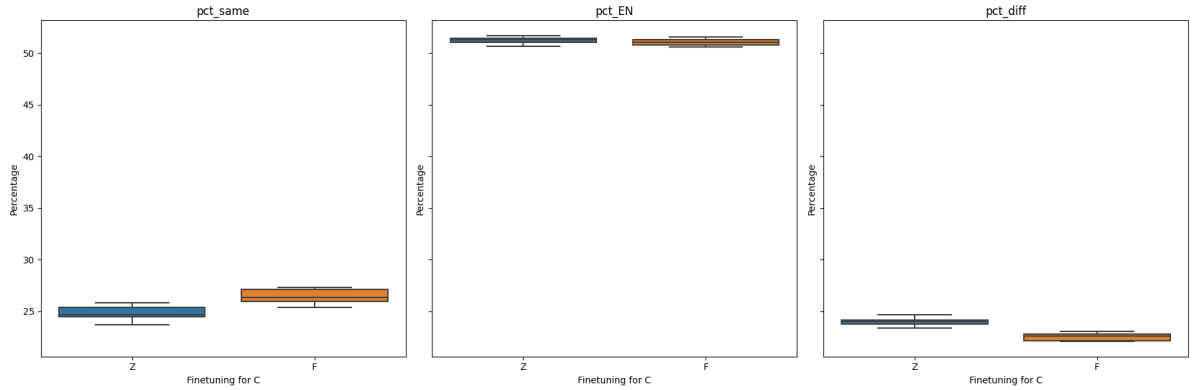


Figure 4.8: Language distribution for zeroshot and fine-tuned ColBERT model

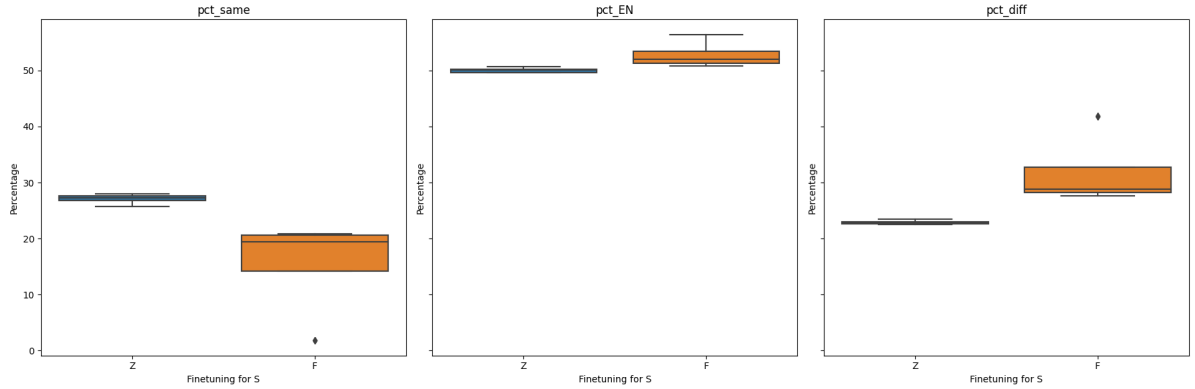


Figure 4.9: Language distribution for zeroshot and fine-tuned SBERT model

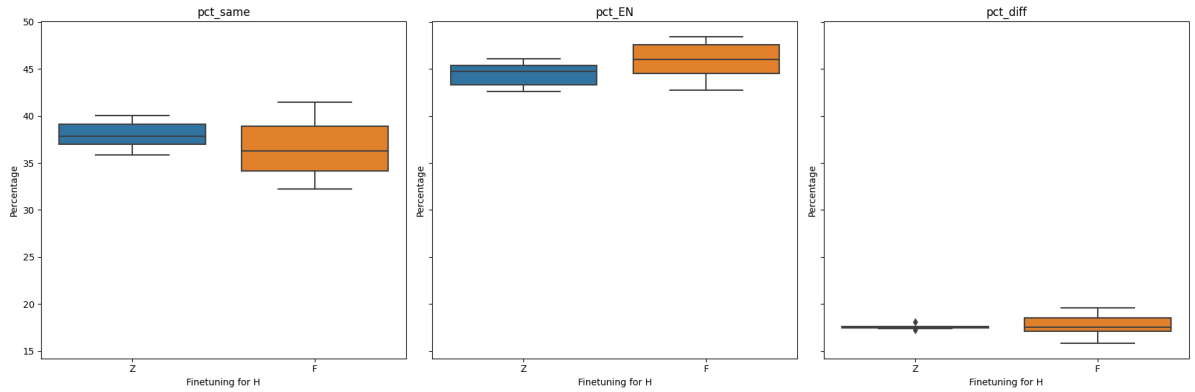


Figure 4.10: Language distribution for zeroshot and fine-tuned Hybrid model

Figures 4.8, 4.9, and 4.10 shows the language distribution over the zeroshot and fine-tuned models for the same, English, and different language document metrics. We can see that the fine-tuning has a noticeable impact with the multilingualism of the retrieval. For Colbert we can see a slight increase in the same-count (monolingualism increasing) and a slight decrease in the different count (multlilingualism decreasing). For Sbert and Hybrid we see larger shifts favouring multilingual retrieval as the same-language count decreases by 9% and 5% respectively and the diff-language count increases by by 7% and 3% respectively. This suggests that fine-tuning enhances multilingual retrieval for SBERT and Hybrid models, allowing them to retrieve more diverse language results, whereas ColBERT maintains a more monolingual bias post-fine-tuning.

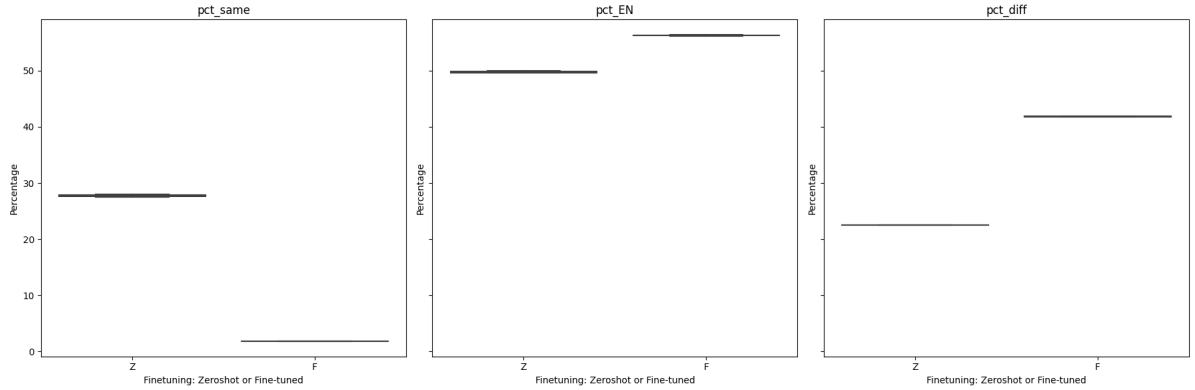


Figure 4.11: Language distribution with respect to finetuning for Sbert with document enrichements only

When analyzing the distribution for each model separately, SBERT exhibits a much larger impact in terms of improved multilingualism, along with significant outliers, as shown in Figure 4.9. Upon faceting by document augmentations, we identify the primary outlier in the case of fine-tuned SBERT with PE (provided + enriched) document data as shown in figure 4.11. This aligns with our findings in the model language distribution, where fine-tuned SBERT with enrichment augmentations led to a notable decrease in same-language retrieval and an increase in cross-language retrieval (as outlined in section 4.3.1).

This behavior could be attributed to SBERT's primary focus on semantic understanding, which may make it more susceptible to overfitting entity-based enrichments compared to other models. When fine-tuned on enriched data, SBERT appears to internalize these entity patterns too strongly, causing a deviation from its usual balanced language distribution. Or it could highlight that something went wrong in the fine-tuning this instance of SBERT as none of the other fine-tuned SBERT systems exhibit this behavior.

Comparative analysis

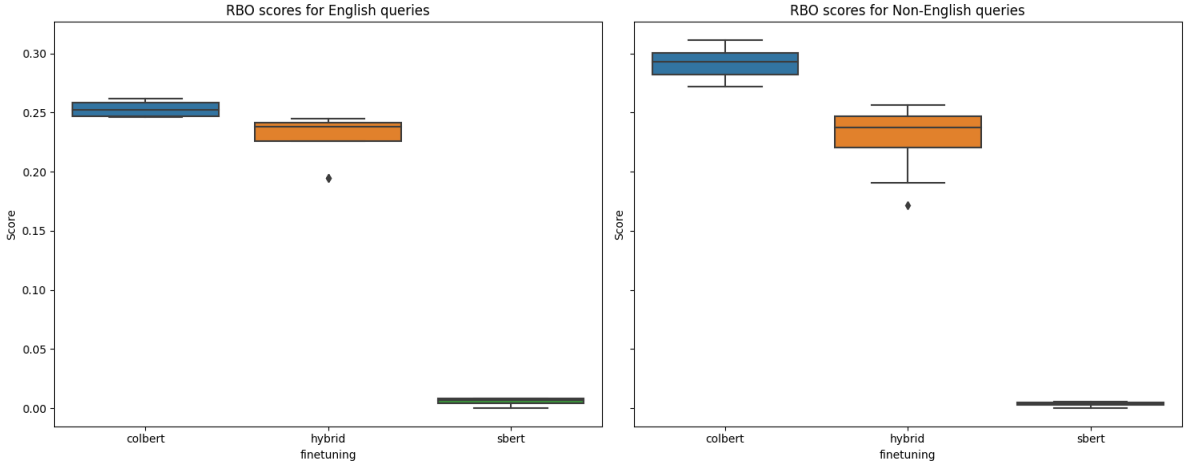


Figure 4.12: Mean RBO scores across all document augmentations for the Hybrid model

The RBO scores in Figure 4.12 measure the similarity between the rankings produced by the zero-shot and fine-tuned variations of the same system, with higher scores indicating greater consistency between the two. A higher RBO score suggests that fine-tuning has a smaller impact on the ranking behavior of the model, while lower scores indicate more substantial shifts.

We observe that the RBO scores are all quite low at approximately 0.25 for English queries and 0.3 for non English queries, for ColBERT and Hybrid. The RBO values significantly lower for SBERT, approaching 0.

This suggests that fine-tuning causes large changes in ranking for all models, with the most dramatic shift occurring in SBERT. The near-0 RBO score for SBERT implies that fine-tuning changes the entire ranking, likely due to its coarse semantic focus, which makes it more susceptible to overfitting to the fine-tuning data.

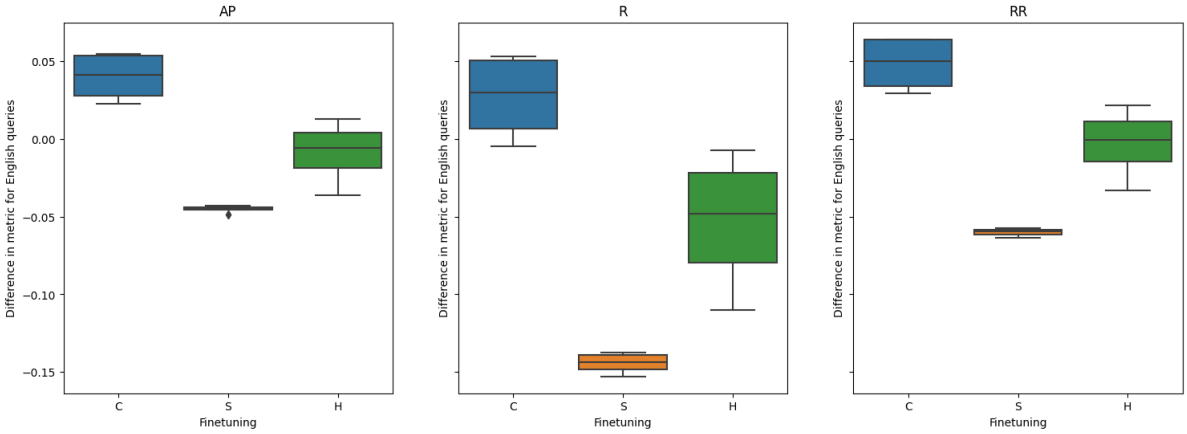


Figure 4.13: Difference in click metrics with respect to the Finetuning for english queries. A negative difference indicates a decline in performance after finetuning, while a positive difference suggests an improvement

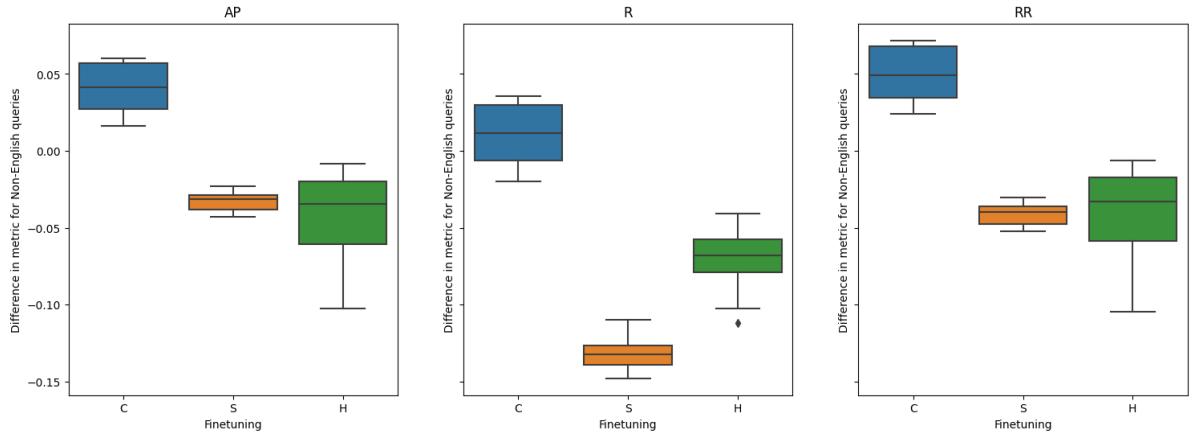


Figure 4.14: Difference in click metrics with respect to the Finetuning for non english queries. A negative difference indicates a decline in performance after finetuning, while a positive difference suggests an improvement

Figure 4.13 and figure 4.14 show us the difference in the scores of model before and after finetuning, with a positive result indicating that the finetuned model has a higher score. We see that for Colbert the finetuning always results in an average increase of 0.05 in the scores for English and Non-English queries. For SBERT we see a constant decrease in scores of 0.15 with all of the metrics suggesting that finetuning actually harms SBERT performance. And for the Hybrid model we see that the average is typically close to 0 with a few outliers that cause the score to increase slightly.

When we look deeper into those outliers for the Hybrid model we see that they are systems which have the full document augmentations (PET). This is shown in figure 4.15 where as the augmentations increase, we can see that the score of the finetuned model also increasing and for the AP and RR, scoring higher than the pretrained model.

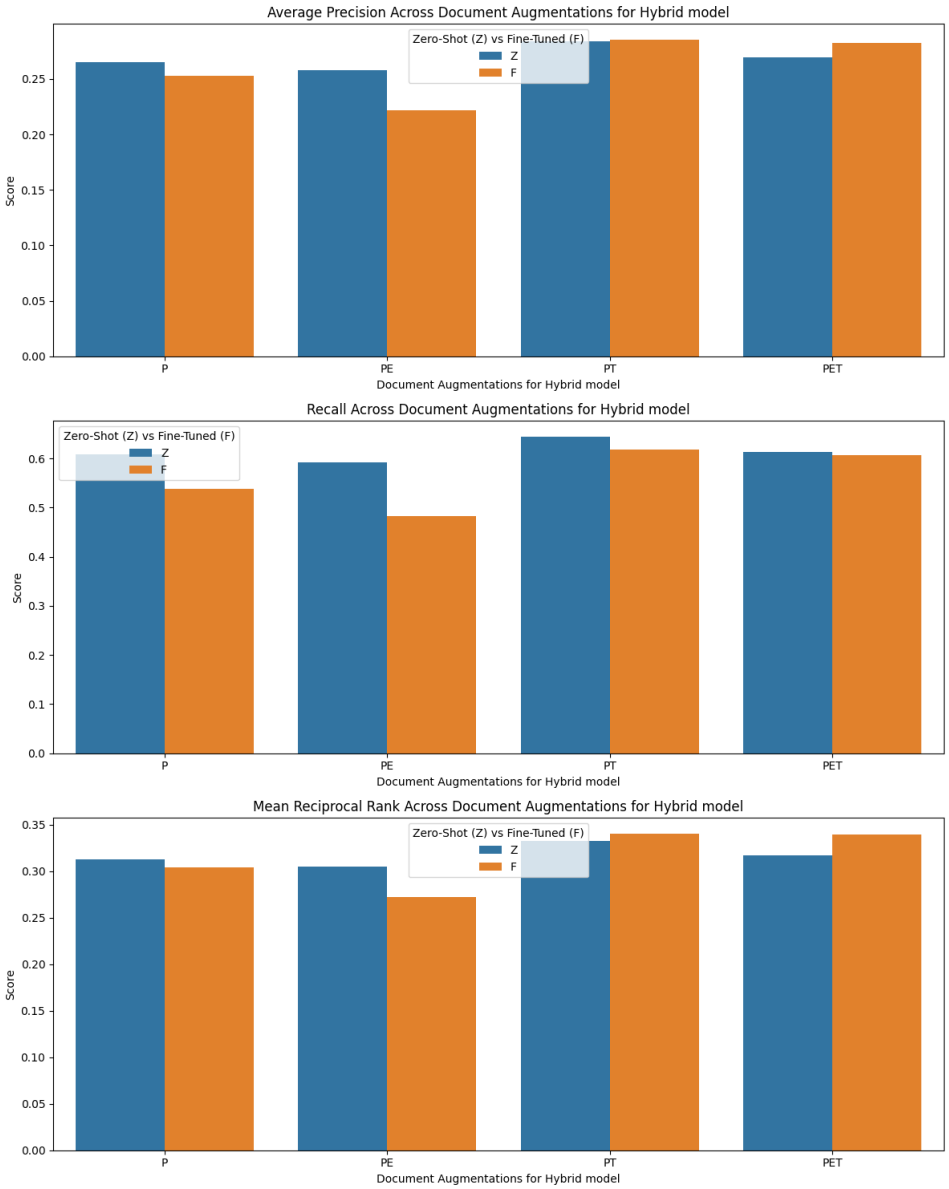


Figure 4.15: Metrics Across Document Augmentations for Hybrid model

4.3.3. Query augmentations

Language distribution

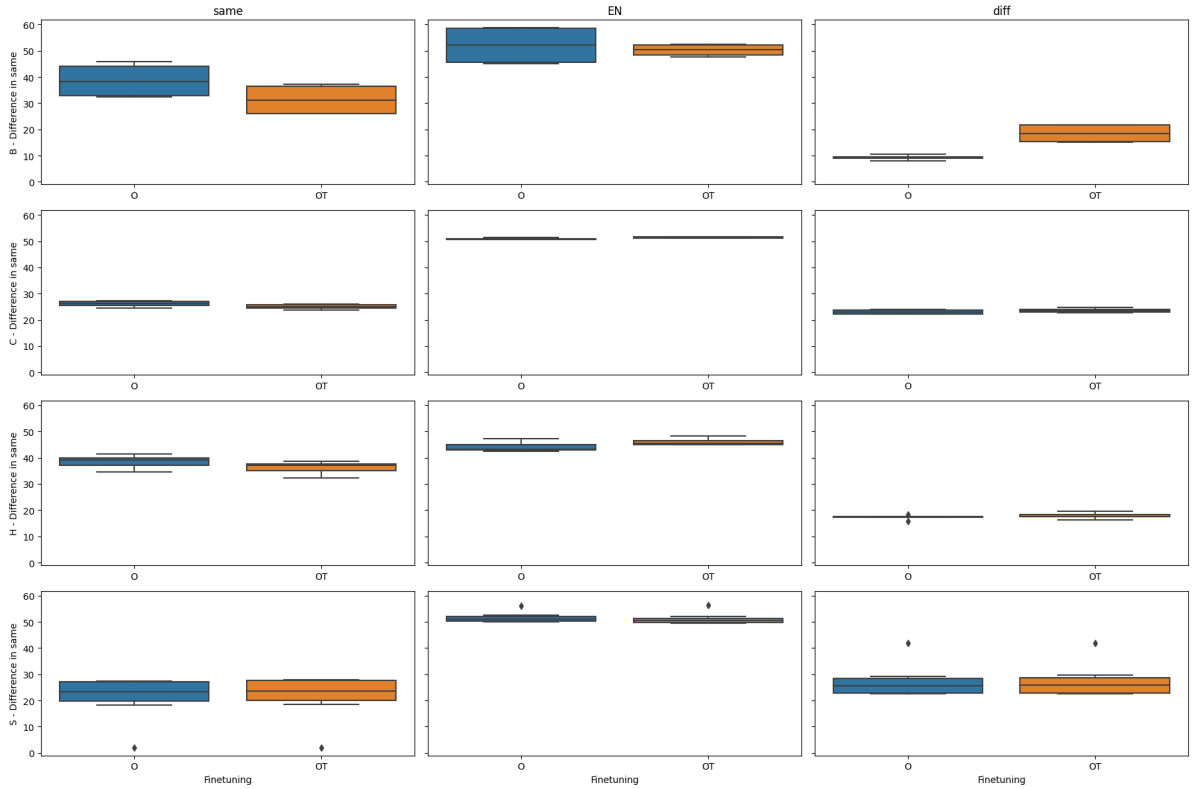


Figure 4.16: Language distribution with respect to Query augmentation

Figure 4.16 shows that query augmentation has minimal impact on language distribution for Neural IR models. ColBERT and SBERT maintain similar distributions (25% same-language, 50% English-language, 25% different-language), while Hybrid remains stable at 40% same-language and English-language, with 20% different-language retrieval. However, BM25 shows a notable shift, with same-language retrieval decreasing by 10% and different-language retrieval increasing by 10%, highlighting its reliance on augmentations for improved multilingual retrieval.

Comparative analysis

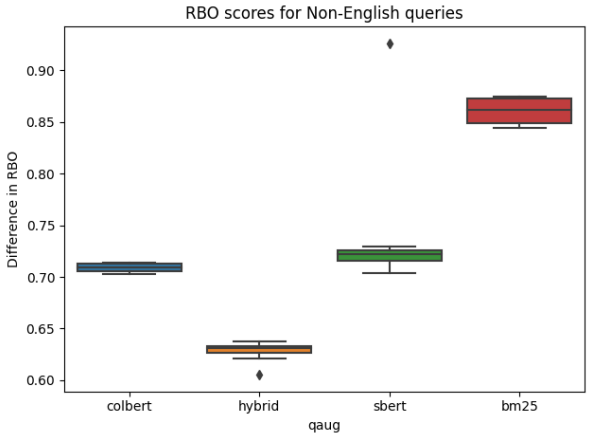


Figure 4.17: RBO with respect to the query augmentations per model for non-english queries only

In Figure 4.17, for non-English queries, we still see relatively high RBO scores, suggesting that translation does influence the rankings, but not drastically. The fact that the rankings remain fairly similar indicates that, while translations introduce some variation, they do not significantly disrupt the overall retrieval patterns. This suggests that the neural models preserve ranking consistency even when handling translated queries, though some shifts occur, possibly due to differences in linguistic structure of the query and document embeddings and semantic interpretation with the translated query.

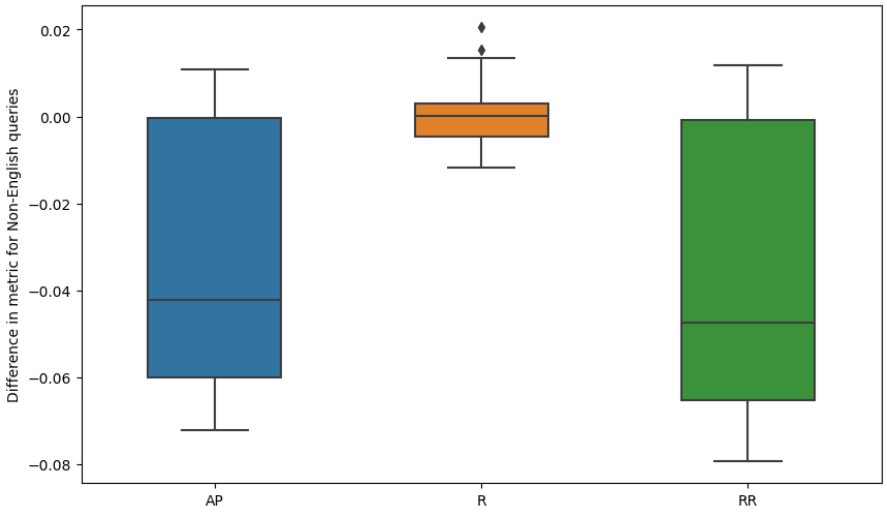


Figure 4.18: Difference in click metrics with respect to the Query augmentation for all models

Examining non-English queries in Figure 4.18, we observe a slight decrease in Average Precision (AP) and Reciprocal Rank (RR) (0.06) and a minor increase in recall (0.02). This suggests that query translation has a limited impact on retrieval effectiveness. The drop in AP and RR may indicate that translations slightly alter document rankings, either pushing relevant results lower or surfacing unjudged relevant documents due to pseudo-judgement bias. Meanwhile, the increase in recall suggests that translation helps retrieve additional relevant documents. Overall, retrieval models remain consistent, with query translation affecting ranking behavior more than overall performance.

4.3.4. Document augmentations: Enrichment's

Language distribution

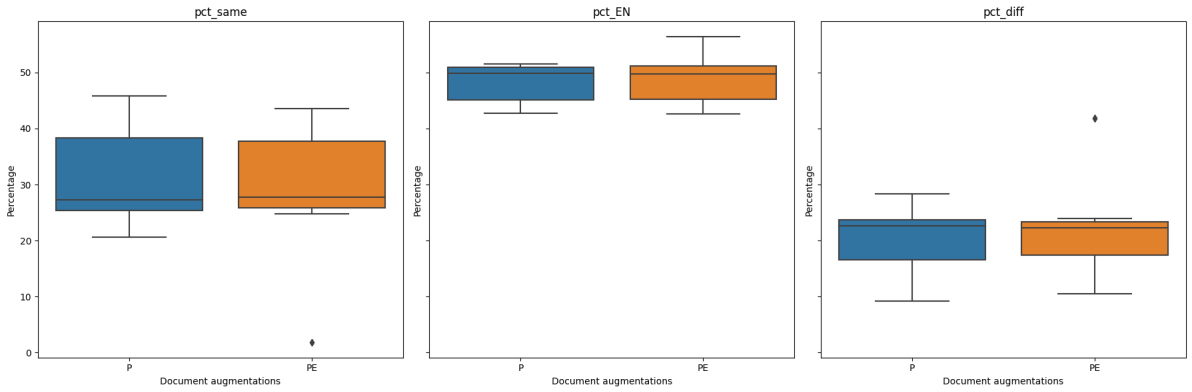


Figure 4.19: Language distribution with respect to the enrichment augmentation: P vs PE

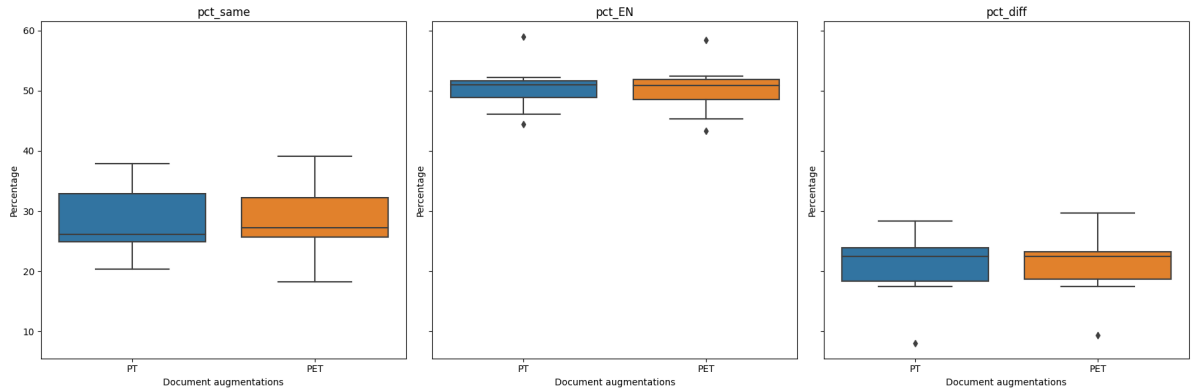


Figure 4.20: Language distribution with respect to the enrichment augmentation: PT vs PET

Figures 4.19 and 4.20 demonstrate that enrichment augmentations have minimal impact on the overall language distributions. Similarly, Figures 4.2a and 4.3 indicate that the enrichment's do not significantly alter retrieval behavior for either BM25 or Neural models.

Comparative analysis

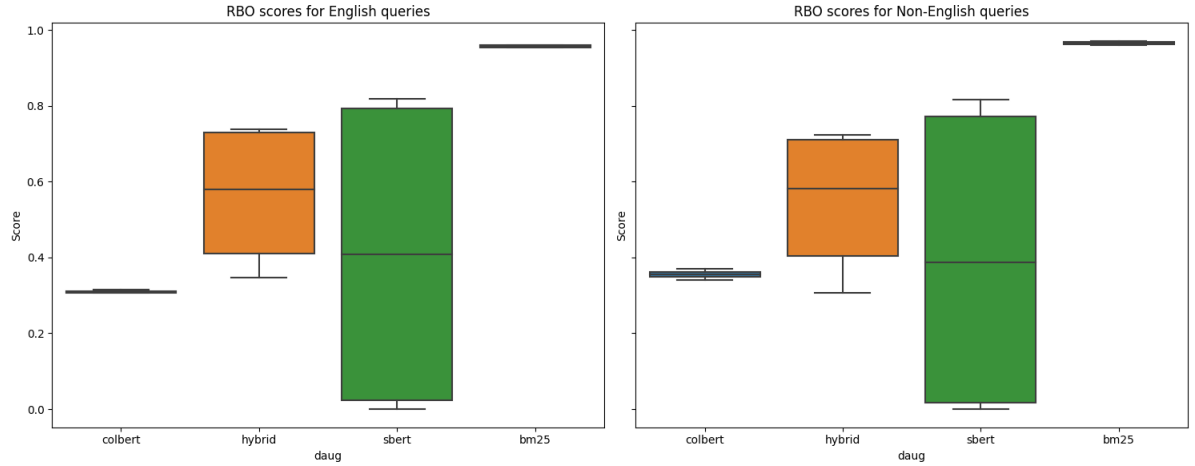


Figure 4.21: Difference in RBO scores with respect to the enrichment augmentation

The RBO scores for document enrichment augmentations show that BM25 remains largely unaffected (0.98–1), indicating near-identical rankings. ColBERT exhibits significant ranking changes, yet its language distribution remains stable (Figure 4.3). Hybrid and SBERT show high RBO scores (0.8) in their pretrained state but drop significantly after fine-tuning, with Hybrid at 0.4 and SBERT nearing 0 (Figure4.22), suggesting fine-tuning amplifies ranking shifts in these models with respect to the document enrichments.

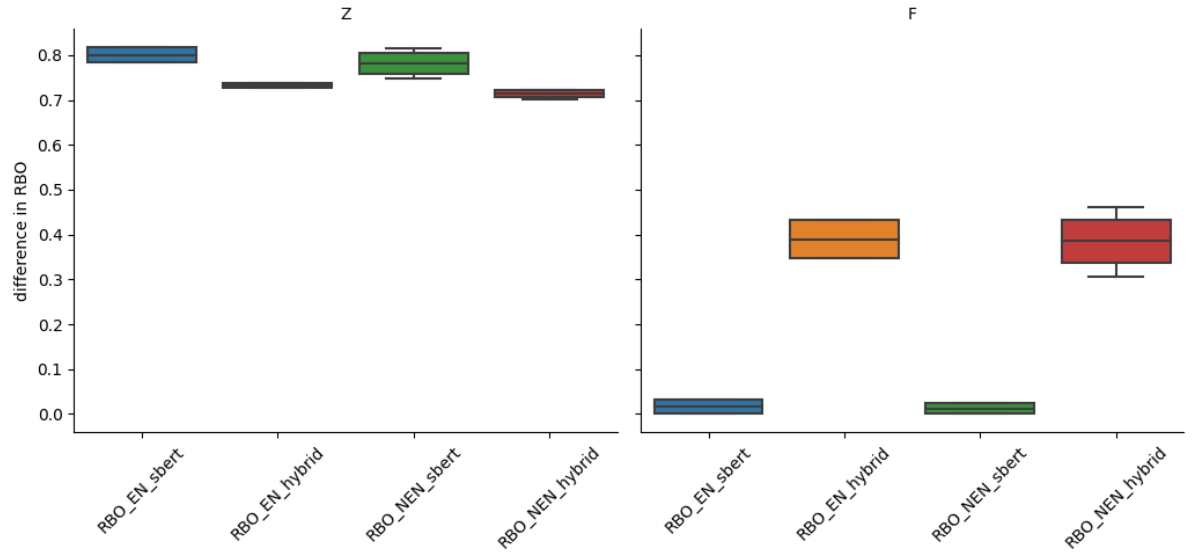


Figure 4.22: Difference in RBO with respect to the enrichment augmentation for zeroshot and finetuned Hybrid and SBERT

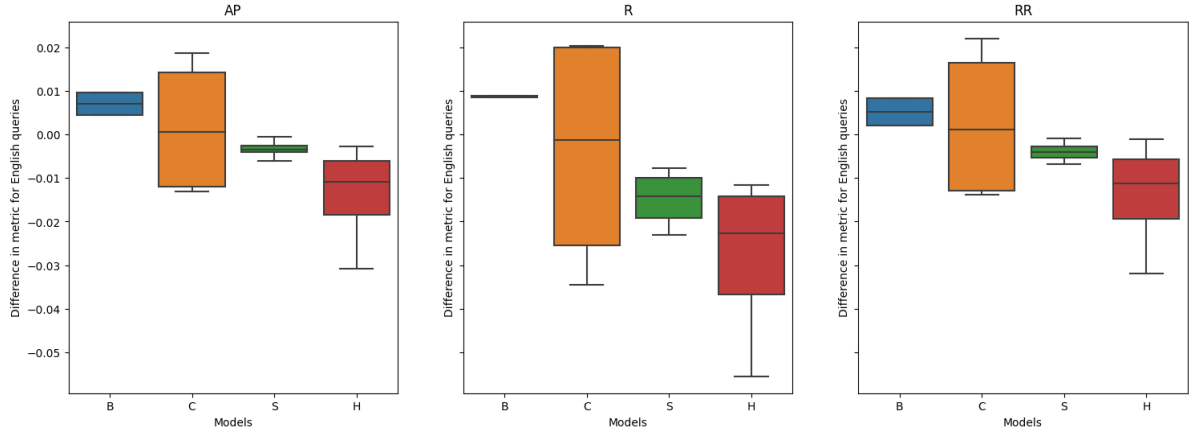


Figure 4.23: Difference in click metrics with respect to the enrichments for english queries

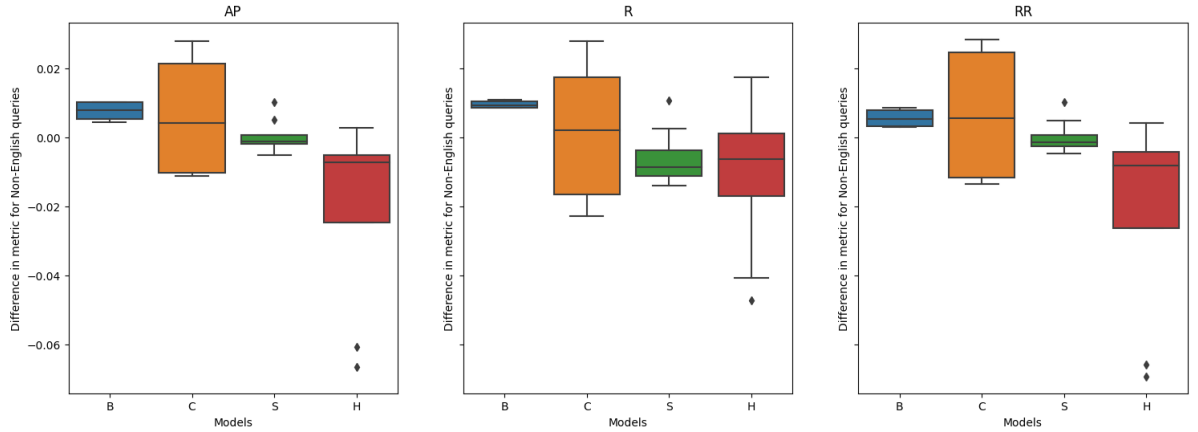


Figure 4.24: Difference in click metrics with respect to the enrichments for non-english queries

The median scores for all metrics, as shown in Figures 4.23 and 4.24, indicate that, on average, the metric

differences are close to 0 across all models. However, ColBERT and Hybrid models exhibit high variability, suggesting that while some systems perform similarly with and without enrichment augmentations, others experience notable improvements or declines in performance. In contrast, SBERT shows a highly stable distribution, indicating that enrichment augmentations have minimal impact on its retrieval effectiveness for all systems. A deeper analysis of ColBERT and Hybrid models reveals a more nuanced trend. Specifically, pretrained ColBERT benefits from enrichment augmentations, achieving higher metric scores, whereas fine-tuned Colbert and Hybrid performs worse when enrichment augmentations are applied. This trend is illustrated in Figure 4.25. Overall the impact of enrichments on performance is not very significant.

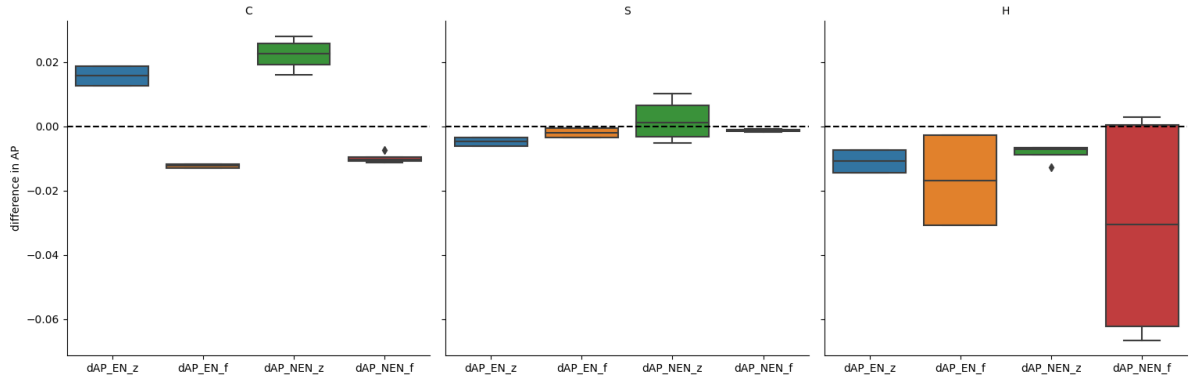


Figure 4.25: Colbert and hybrid scores separated by pretrained and finetuned versions

4.3.5. Document augmentations: Translations

Language distribution

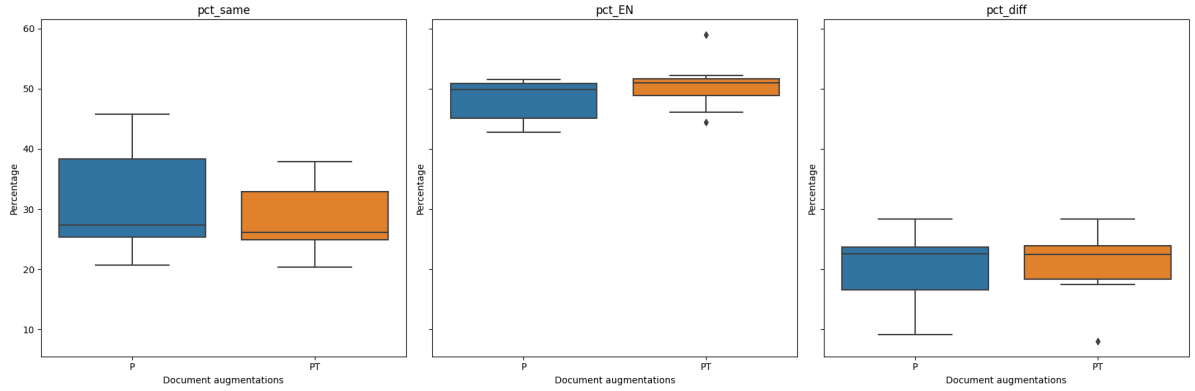


Figure 4.26: Language distribution with respect to the translation augmentation: P vs PT

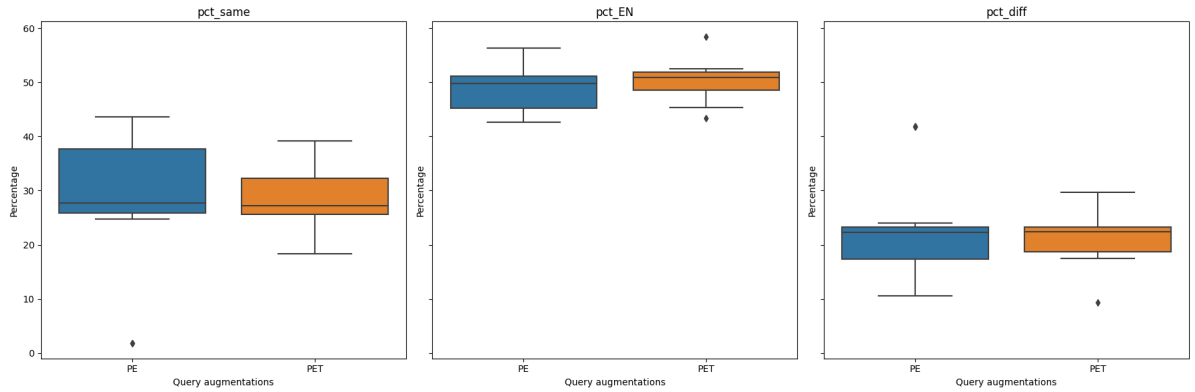


Figure 4.27: Language distribution with respect to the translation augmentation: PE vs PET

The language distribution with respect to the translation does not change drastically, indicating the distribution of retrieved document languages remains stable across different augmentation strategies. However, there are much higher outliers present for the english count and when looking deeper we see that this is for the BM25 model which has a significant rise in english document count. The neural models are very stable and implies that they effectively integrate document translations.

Comparative analysis

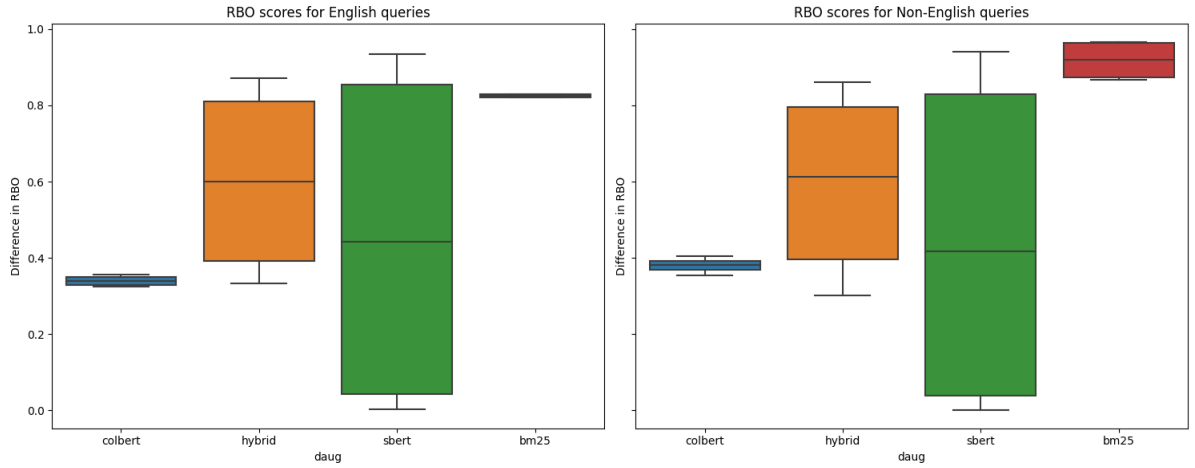


Figure 4.28: Difference in RBO scores with respect to the Translation augmentation

When looking at the RBO scores with respect to the document translation augmentations in figure 4.28, we can see that Bm25 is high and Colbert is low as with the other augmentations indicating that BM25 models rank similarly and the difference in rankings are the translations heping the model find english documents based on the translation. Colbert has a low score, as seen with the previous augmentations. For Hybrid and SBERT we notice a very high range - looking deeper we find out that this range is split between the finetuned and petrained models. Pretrained models have high RBO scores for Hybrid and SBERT and fientuend models have low scores. Colbert is the same regardless of finetuning.

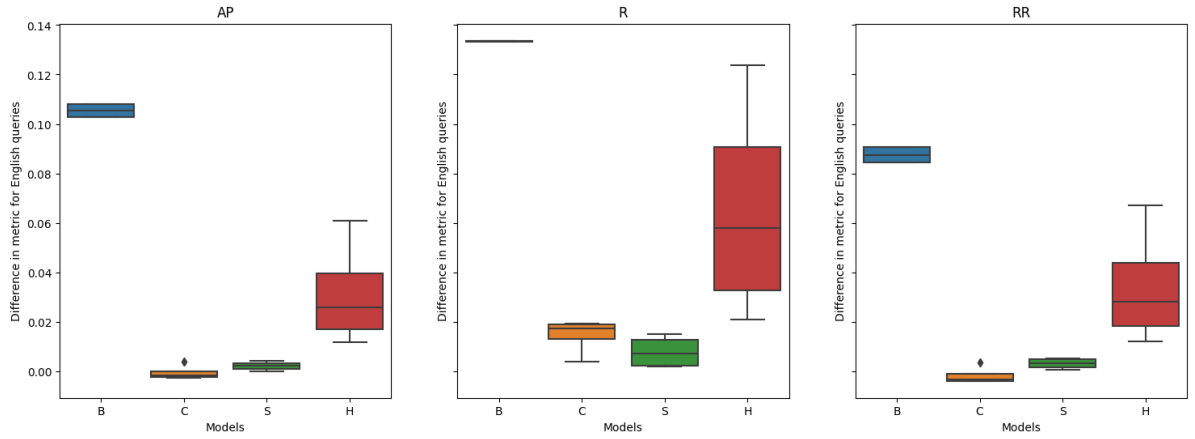


Figure 4.29: Difference in metrics with respect to the Translation augmentation for english queries

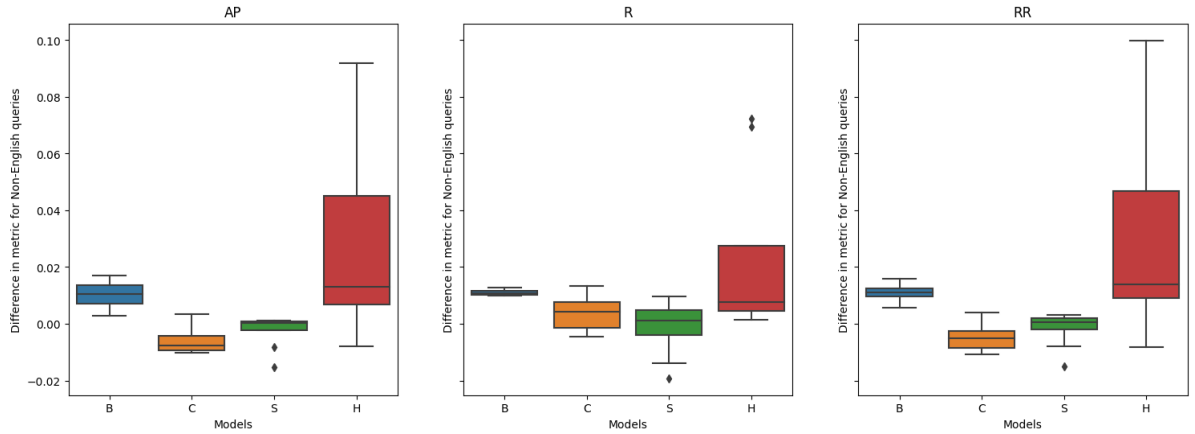


Figure 4.30: Difference in metrics with respect to the Translation augmentation for non-english queries

Based on Figures 4.29 and 4.30 we observe that, on average, the difference in scores for non-English queries is 0 or positive across all three metrics. While the magnitude of the improvement is very small, this trend suggests that document translations contribute to a modest but consistent enhancement in retrieval performance.

4.4. Ranked list truncation

One of the key challenges encountered during the evaluation was the presence of numerous low-quality results in the retrieved ranked lists. This issue arises because Neural Information Retrieval (NIR) models, by design, assign scores to all documents in the collection and retrieve the top k results based on those scores, regardless of their actual relevance. As a result, the lower ranks of the retrieved lists often contain documents with extremely low scores, which contribute little to the evaluation and may dilute meaningful insights. We observed this in the previous section, where for some augmentations the rankings would change substantially depending on the augmentations applied and the RBO values would be quite low, but the overall effectiveness metrics would remain the same or exhibit only marginal changes. This means that those augmentations would cause the documents at the top have changed considerably.This suggests that a significant portion of the results—particularly those in the lower ranks—are irrelevant or minimally impactful in terms of the system's retrieval performance. These irrelevant documents, often included due to augmentations introducing additional lexical or semantic matches, not only add noise to the ranked lists but also make meaningful comparisons between systems more challenging.

To better understand this issue, we analyzed the score distributions in the ranked lists and plotted the

scores across the retrieved ranks. The plot reveals that scores for many models exhibit a sharp decline initially and then plateau at low values. This pattern indicates that a significant portion of the retrieved results consists of documents with poor semantic relevance to the query. While this phenomenon is more pronounced in NIR models, it is also observed in BM25 for certain queries, likely due to scoring artifacts caused by the presence of stopwords, subwords, or terms that are only marginally related to the query context.

4.4.1. Method

Based on our observations, we propose dynamically truncating the ranked lists to exclude the low-quality results that contribute little to the evaluation. By analyzing the score curves, we identified the "curve point" where the scores begin to plateau. This point serves as a natural cutoff, allowing us to retain only the results with meaningful scores.

To implement this, we used a simple heuristic approach:

- Compute the scores for all documents in the ranked list.
- Fit a curve to the score distribution to identify the inflection point where the slope changes significantly.
- Use this inflection point as the cutoff rank, truncating the list at this point.

This method dynamically determines the truncation threshold for each query, ensuring that only the most relevant results are retained while eliminating noise from the evaluation. For consistency, the same process was applied to all retrieval models, including BM25 and neural models.

4.4.2. Truncated results

After applying the ranked list truncation process, we analyzed the impact on retrieval effectiveness and language distributions. Overall, we observe that while the truncation has a notable effect on language distributions, it does not substantially alter the rank similarity (RBO) or retrieval performance metrics across models.

The language distribution analysis of the truncated results reveals shifts due to query and document augmentations, particularly for BM25 and Hybrid models. This suggests that truncation helps isolate the most relevant retrieved documents, allowing a clearer view of how different augmentation strategies influence retrieval. Specifically, the truncation reduces noise introduced by irrelevant or low-score documents, making it easier to observe systematic retrieval trends across different models.

However, despite these changes in language distribution, the overall rank similarity (RBO) and retrieval metrics (AP, RR, Recall, etc.) remain relatively stable before and after truncation. This is expected because RBO results weigh the top ranks more. Thus, the fundamental ranking behavior of the models is preserved, reinforcing the idea that only the highest-ranked documents contribute meaningfully to user experience and evaluation outcomes.

From these findings, we argue that truncating results provides a more realistic view of model performance in practical search scenarios, where users typically focus on the first few retrieved documents. By filtering out low-relevance results, truncation enhances our ability to assess how augmentations influence retrieval effectiveness, particularly in multilingual settings, without introducing unnecessary noise into the evaluation.

4.5. Qualitative analysis

While our quantitative evaluation provides, to some extent, insights into the behavior of the retrieval models under different configurations, it is important to acknowledge that our evaluation is based on biased pseudo-judgements towards BM25. As a result, while the trends we observe in the metrics are informative, they are not necessarily fully reliable indicators of true retrieval effectiveness.

To address this limitation, a qualitative analysis is necessary to examine specific cases where retrieval behavior diverges significantly. This allows us to assess whether the neural models are genuinely underperforming or if their retrieval strategies simply do not align with our pseudo-judgement-based evaluation.

For each analysis, we identify a set of queries aligned with our focus and randomly sample 3-4 for manual relevance annotation. We then compute their AP scores to assess retrieval effectiveness. Additionally, we examine the retrieved documents to hypothesize about model behavior, recognizing that with a limited qualitative sample, we cannot definitively prove underlying retrieval patterns but can gain valuable insights.

4.5.1. BM25 vs Neural models

Our results show that neural models score lower on the metrics, raising the question of whether this reflects worse retrieval or if they retrieve relevant documents overlooked by our biased evaluation. Examining selected queries will help clarify their true effectiveness.

For this comparison we looked into the queries which had high AP scores for BM25 and low scores for the neural models. We only looked into the models with full augmentations, as this represented the 'most multilingual' BM25; so all models with full query and document augmentations (<model>-OT-PET). We then obtained the following queries - the information in the parenthesis are not part of the query but to provide context:

- Latvian national archive
- grimming (refers to a mountain in Austria)
- escultura en grecia, Translated: sculpture in greece
- benvenuto cellini, Translated: welcome cellini (Benvenuto Cellini was an Italian sculptor)

After judging the results of these models for these queries ourselves we obtained the following results: When looking at the overall scores among the models in table 4.8, we see that the neural models actually

System	MAP
B-OT-PET	0.6698
C-OT-PET	0.7500
H-OT-PET	0.7500
S-OT-PET	0.1698

Table 4.7: Model comparison based on MAP and MRR scores.

perform better across these selected queries based on our judgements.

Query	System	Score: AP
Latvian national archive	B-OT-PET	0.0000
	C-OT-PET	1.0000
	H-OT-PET	1.0000
	S-OT-PET	0.0000
Original: benvenuto cellini,Translated: welcome cellini	B-OT-PET	1.0000
	C-OT-PET	1.0000
	H-OT-PET	1.0000
	S-OT-PET	0.0000
Original: escultura en grecia,Translated: sculpture in greece	B-OT-PET	0.6792
	C-OT-PET	1.0000
	H-OT-PET	1.0000
	S-OT-PET	0.6792
grimming	B-OT-PET	1.0000
	C-OT-PET	0.0000
	H-OT-PET	0.0000
	S-OT-PET	0.0000

Table 4.8: Model performance for selected queries across different systems.

However when looking closely at the retrieved results and at the scores for each model for the respective queries in table 4.8, we can see some differences across the queries for the scores and can infer some interesting behavior about the models' retrieval.

For the Latvian national archive query, Hybrid performs well, retrieving highly relevant documents. ColBERT, while finding one correct document, also ranks many Lithuanian archives highly, likely due to the presence of the language tag "lat" in metadata fields, causing it to associate them incorrectly. SBERT behaves similarly to ColBERT but exhibits even greater misalignment in terms of country and institution, making all results irrelevant. BM25, however, fails entirely, returning results from the French national archive, suggesting that its strict reliance on term matching leads to incorrect associations and it not being able to understand the user' intent as they requested for documents from a specific source.

For grimming, a mountain in Austria, BM25 successfully retrieves relevant documents, whereas all neural models fail. This may be due to tokenization effects, where the models misinterpret or associate the term "Grimming" with other tokens like "Grym" or "Grim." This highlights a potential weakness in neural models for handling singular entity queries, where token-based confusion may degrade retrieval effectiveness.

For escultura en grecia ("sculpture in Greece"), most models retrieve relevant results, though some include Cypriot sculptures. BM25 likely retrieves Cypriot sculptures when the description explicitly mentions Greece, whereas neural models may interpolate between Greece and Cyprus based on semantic similarities. This suggests that neural models are influenced by implicit contextual relationships, whereas BM25 follows a more explicit term-matching strategy.

For Benvenuto Cellini, the translation to "Welcome Cellini" is misleading. BM25 performs well, as the original name still matches, but SBERT suffers the most, likely due to its sentence-level embeddings misinterpreting the phrase. In contrast, ColBERT and Hybrid remain relatively robust, likely because their token-level and lexical representations respectively allow them to disregard the incorrect translation.

These observations suggest that neural models struggle with "singular entity queries", likely due to tokenization effects, whereas BM25 struggles with "user intent" and may rely too heavily on exact term matching. Additionally, language tags may, which we include in the data, introduce noise for neural models while aiding BM25, raising the question of whether they should be included in retrieval. Overall, this qualitative analysis highlights the strengths and weaknesses of each approach, reinforcing the need for evaluation beyond pseudo-judgements.

4.5.2. SBERT with enrichments and fine-tuning

Our results show that SBERT is highly sensitive to enrichments and fine-tuning, leading to drastic shifts in language distribution and inconsistent retrieval performance. Unlike ColBERT and Hybrid models, SBERT overfits to entity patterns and metadata, struggling to generalize across multilingual content. Surprisingly, SBERT's retrieval often declines after fine-tuning, suggesting it adapts too narrowly to training data.

To explore this, we analyze queries where SBERT's rankings change significantly with enrichments and fine-tuning, assessing whether these shifts improve retrieval or indicate overfitting and loss of relevance diversity. We found the following queries:

- altonaer nachrichten 16.10.1933 (A German newspaper)
- Sponge cake
- History of coffee

After judging these queries' results before and after enrichment and fine-tuning for SBERT, we see the following results.

From these results, we observe that SBERT with fine-tuning performs significantly worse, with MAP scores dropping to 0.0000 when combined with enrichment data (SF-O-PE and SF-OT-PE). This suggests either an issue with the fine-tuning process or that SBERT is inherently poor at leveraging enriched metadata. In contrast, SBERT with fine-tuning and without enrichments (SF-O-P and SF-OT-P) maintains non-zero scores indicating that within the top 5 its still able to find results.

System	MAP
SF-O-P	0.2778
SF-O-PE	0.0000
SF-OT-P	0.2685
SF-OT-PE	0.0000
SZ-O-P	0.8514
SZ-O-PE	0.8444
SZ-OT-P	0.9444
SZ-OT-PE	0.9185

Table 4.9: MAP scores for different system configurations.

When analyzing the results and scores per query, we observed no major differences across queries. However, a closer examination of the retrieved documents revealed interesting retrieval patterns. For Altonaer Nachrichten 16.10.1933, SBERT frequently retrieved other historical documents from the same period, likely due to the prominence of the date in metadata. This suggests that SBERT model might prioritize the occurrence of metadata promoting a document if something in it is repeated a lot, even if they are not completely relevant. For Sponge cake, many results were about "sponge animals" rather than the intended culinary term, highlighting SBERT's difficulty with disambiguation in enriched settings.

Overall these findings suggest that enrichments do cause a significant downgrade but only when finetuned. Possibly indicating that the fine-tuning process for SBERT did not go well and must be re-evaluated.

4.5.3. Zeroshot vs Finetuned models

fine-tuning significantly impacts retrieval performance, but its effects vary across models. To investigate, we compare zeroshot and finetuned models on key queries, assessing whether fine-tuning improves relevance or introduces retrieval errors.

We find queries where the RBO scores between the zeroshot and finetuned models are low, indicating that the ranks are very different. We want to see what impact this has on the retrieval of the documents. For this analysis we randomly obtained these queries:

- jacob van hulsdonck (Flemish painter)
- mulroy bay (place in Ireland)
- the girls of slender means (A book by Muriel Spark)

The queries we obtained from this selection are all entities. We then calculate the AP scores for the retrieved documents per query: We see that for these specific queries, which are all entities, the AP

System	MAP
CF-OT-PET	0.5111
CZ-OT-PET	1.0000
HF-OT-PET	0.7685
HZ-OT-PET	0.8056
SF-OT-PET	0.0000
SZ-OT-PET	0.2958

Table 4.10: Mean Average Precision (MAP) scores for different system configurations.

scores of the finetuned models decrease. This suggests that finetuned neural models may not be the most effective at retrieving single-entity queries, as they likely rely more on contextual cues rather than exact entity matches. For isolated entity queries, the lack of additional context may cause neural models to misinterpret the intent or associate the entity with semantically related but non-relevant documents.

This is evident in the retrieved results for the finetuned models, where instead of retrieving the exact entity, they return documents related to “slender girls” for The Girls of Slender Means and other bays (for example Hudson Bay) for Mulroy Bay.

When looking at the models we can see that the difference in scores between zeroshot and finetuned colbert is quite significant but the difference between zeroshot and finetuned Hybrid is much smaller. This could be because of the lexical retrieval of the hybrid model which helps keep retrieval to exact matches, making it less sensitive to fine-tuning adjustments for entity queries. In contrast, ColBERT’s token-level interactions are more susceptible to embedding shifts, causing greater variability in rankings after fine-tuning.

This suggests that the dense embedding models rely more on broader semantic associations. One possible explanation is that the document data does not provide strong entity-specific signals, leading the models to favor conceptually related content over exact matches. fine-tuning, which optimizes models for domain-specific patterns, might further reinforce this bias toward contextual relevance over strict entity recognition, reducing retrieval effectiveness for entity-based queries, highlighting potential limitations with neural models.

5

Discussion

5.1. Effectiveness: Results Discussion

Based on our quantitative and qualitative results we can gauge the impact of each treatment to the system and how they impact performance, helping us answer the first sub-question:
How do different Neural IR models and treatments—particularly the use of translation, enrichment stages, and fine-tuning on Europeana's dataset—impact retrieval performance compared to the BM25-based approach?

5.1.1. Model

From the language distribution analysis, we can observe that NIR models, particularly Hybrid and ColBERT, show more balanced language distribution compared to BM25. This balanced distribution of the neural models suggests that they are less impacted by the augmentations and are better equipped to handle multilingual retrieval scenarios naturally. For BM25 we can observe that the query and document augmentations do have a significant impact on the language distribution as they help expand the search beyond same-language matches. The translations enable better matching with English document, suggesting that augmentations are important to broaden BM25's multilingualism. T

For the RBO and performance metrics we see that there is a huge difference in rankings between the models which causes a negative impact on the perofrmance scores. This is largely influneced by the pseudo-judgements we use. This biased-evaluation suggests that the neural models always perform worse than the BM25 models. However in our qualitative analysis we were able to demonstrate that the neural models were actually able to perform better than the BM25 and retrieve more relevant documents, demonstrating their strength in semantic matching. Suggesting that our quantitative analysis does not correctly capture the strength of the neural models.

Overall, while neural IR models may be more well suited for multilingual retrieval scenarios, careful consideration must be given to conduct deeper evaluation that can better capture their actual performance benefits.

5.1.2. Fine-tuning

From the language distribution analysis we can see the individual impact of fine-tuning for each model. Since ColBERT independently compares query and document token embeddings, it is already well-suited for multilingual retrieval, reducing the need for additional fine-tuning to capture language-specific nuances. This could explain why fine-tuning has little effect on ColBERT's language distribution. In contrast, Hybrid models exhibit more sensitivity to fine-tuning in terms of language distribution. This increased variability likely stems from the interaction between lexical matching and semantic representations, where fine-tuning can shift the balance between these two mechanisms. As a result, Hybrid models are more prone to language fluctuations based on fine-tuning.

For SBERT, fine-tuning has a strong impact on ranking distributions, but unexpected behavior emerges

when documents are enriched. SBERT primarily relies on sentence-level embeddings, which capture semantic similarity but lack fine-grained control over individual tokens. When exposed to enriched data, it appears to overfit to entity patterns and linguistic structures present in the fine-tuning set, leading to instability in language distribution. Our qualitative analysis supports this observation, showing that fine-tuned SBERT performs poorly on enriched documents. While this suggests that entity-rich augmentations negatively impact retrieval, it remains unclear whether the root cause is inherent model sensitivity to entities or potential issues in the fine-tuning process. Further investigation is needed to disentangle these factors. As observed in Zeroshot vs Finetuned qualitative analysis, fine-tuning has the potential to improve retrieval, suggesting that the negative impact seen with SBERT is likely due either a mistake in finetuning that system or an inherent sensitivity to enrichments. Further investigation is needed.

Fine-tuning the models causes significant shifts in document rankings, as seen in the consistently low RBO scores. This could be due to fine-tuning adjusting vector-level matching to Europeana's data, changing which words and passages the models prioritize. These shifts are influenced by model architecture, with ColBERT focusing on token-level interactions, SBERT on sentence-level embeddings, and Hybrid models balancing lexical and dense retrieval. Additionally, our document formatting choices for NIR, including structuring metadata as free text with the field names (outlined in section 3.2.1), may have contributed to these ranking changes. The way the documents were structured could have amplified certain retrieval biases, causing fine-tuned models to diverge significantly from their pretrained rankings, aligning more with domain-specific patterns and user interactions.

When analyzing the performance, we observe that ColBERT and Hybrid models benefit from fine-tuning under certain conditions, leading to improved performance. ColBERT shows consistent performance improvements while maintaining language stability, regardless of augmentations, whereas Hybrid benefits most from full document augmentations. However, SBERT does not exhibit any gains and shows significant loss in performance, indicating that the finetuning was not done properly.

Overall, fine-tuning significantly impacts document rankings and retrieval performance, but its effectiveness varies across models. Additionally, document formatting choices may have influenced ranking shifts, amplifying retrieval biases. These findings highlight that fine-tuning must be carefully tailored to each model's architecture and data augmentation strategy to achieve optimal retrieval in Europeana's search system.

5.1.3. Query augmentation

The query augmentations have minimal impact on the inherent multilingual capabilities of neural IR models, while providing some benefit to BM25 through translations. This is also seen in the RBO's which, compared to other treatments, are the highest indicating that the ranks do not change much with the inclusion of translated queries.

The effectiveness varies by model architecture, with SBERT showing slight improvements in scores due to enhanced semantic understanding in the query, while ColBERT and Hybrid models experience minor degradation in scores due to potential query noise from less precise translations as seen in the qualitative analysis; some translations were wrong and misleading the neural models.

This suggests that query augmentations is only valuable for traditional retrieval methods and doesnt impact the neural models as much.

5.1.4. Document augmentations

Enrichments

In terms of language distribution, enrichment's do not introduce any substantial changes across neural models. However it does slightly improve the multilingual distribution for BM25 as shown in figure 4.2a.

For pretrained neural models, enrichment's offer minor benefits, with ColBERT demonstrating the most consistent improvements. The fine-grained token-level matching of ColBERT allows it to leverage enriched metadata effectively, leading to better retrieval performance in some cases. However, once finetuned, ColBERT's performance declines when only-enriched data is introduced, suggesting that for

enrichment's only, the model starts to falter. This was also shown in our qualitative analysis where we argued neural models might struggle with handling entities.

Overall, enrichments slightly improve BM25's multilingual retrieval but have very minimal impact on Neural IR models, both in terms of language distribution and performance. The language distribution does not really change and the retrieval slightly decreases; probably due to entity based enrichments not doing well with neural models.

Translations
For BM25, translations enhance multilingual retrieval capabilities, making it more effective at retrieving documents in different languages. This improvement occurs because translated documents introduce consistent English representations across the dataset, allowing BM25 to match to more terms.

For Neural IR models, the influence of translations is more nuanced. ColBERT and Hybrid models show minimal changes in language distribution, suggesting that multilingual embeddings and dense representations already account for cross-lingual variations, reducing the need for additional translation-based augmentation. However, SBERT exhibits improved stability when transitioning from enriched to enriched-translated settings (PE $\rightarrow$ PET), indicating that translations help reinforce semantic consistency in its sentence-level embeddings. This suggests that SBERT, which lacks fine-grained interaction mechanisms, benefits more when translations create uniform representations across languages, helping to mitigate inconsistencies caused by multilingual variation.

Overall, translation-based document augmentations do not have a negative impact across models. The effect is either neutral or positive, but on its own, it remains limited. While translations introduce English-aligned representations that could aid cross-lingual matching, their benefit depends on the model's ability to leverage them effectively. In our results, improvements in metrics are minimal, suggesting that translations alone do not significantly enhance retrieval performance for most neural models.

5.1.5. Other observations
Something we notice throughout the various configurations and augmentations is the sensitivity of ColBERT. For all of the aforementioned configurations we see that when the respective change is applied, either to the zeroshot or fintuned models, the RBO scores are always indicating that no matter what change is made ColBERT rankings change drastically. This could be because of how fine-grained the model is. Even minor modifications, such as reformatting passages, altering sentence boundaries, or changing tokenization, it changes where and how tokens are stored in ColBERT's per-token embedding index. Consequently, any structural adjustments affect how query tokens interact with document tokens, potentially amplifying ranking changes, even when the underlying content remains the same.

We notice this further with document augmentations, where SBERT and Hybrid models exhibit high RBO scores with respect to the document enrichments and translations, before fine-tuning but experience a significant drop afterward. The consistently low RBO scores for ColBERT, even in its pretrained state, contrast with the behavior of SBERT and Hybrid models and can be attributed to ColBERT's fine-grained token-level interaction mechanism.

In contrast, SBERT and Hybrid models initially have high RBO scores with respect to enrichments and translation, before fine-tuning, suggesting that their pretrained ranking behavior remains relatively stable as document augmentations increase. This stability arises because SBERT processes information at the sentence level, creating dense embeddings that capture overall semantic similarity, rather than focusing on specific token interactions. Similarly, Hybrid models balance lexical and dense retrieval mechanisms, which mitigates the effect of minor textual modifications in the absence of fine-tuning. However, after fine-tuning, these models become more sensitive to enriched and translated content, leading to a drop in RBO scores. This is agani probably due to the finetuning changing the way the models represent embeddings, changing the similarity space and leading to shifts in document rankings. We see in the qualitative analysis for zeroshot vs finetuning that this might result in a deterioration in performance.

5.2. Efficiency: Infrastructure considerations

Based on our investigation, we have identified key hardware requirements and efficiency constraints for running Neural IR models at scale, helping us answer the second sub-question:
What are the infrastructural and efficiency considerations for implementing Neural IR in Europeana?

Our experiments on Europeana's RND-3 server highlighted severe computational limitations, while tests on DAIC provided insights into the resources needed for scalable indexing and retrieval.

One of the most critical limitations in our experiments on RND-3 was the GPU hardware. The NVIDIA GTX 1080 (8GB VRAM, Pascal architecture) GPU could not run the Jina-ColBERT v2 model. This showed us that to support modern Neural IR approaches, Europeana requires high-memory GPUs, such as the NVIDIA A40 (48GB VRAM) or A100 (40GB/80GB VRAM) GPU's available on DAIC.

CPU and RAM constraints also impacted indexing performance. The Intel Core i7-7700 (8 cores) and 62GB RAM on RND-3 proved inadequate for batch indexing and parallel document processing. Since NIR models require high-memory vector indexing, insufficient RAM led to slow indexing times and potential crashes. A more suitable configuration would include a system with at least 256GB–512GB RAM, ensuring efficient batch processing and dense vector storage in memory.

Storage and scalability are also key considerations. Traditional BM25-based inverted indexes are lightweight, but ColBERT and Hybrid models require significantly more storage. Our experiments revealed that indexing 1 million documents required up to 10GB for ColBERT and 8GB for Hybrid models. To scale indexing effectively, Europeana should look for a storage system of around 1TB or use cloud-based solutions.

As for indexing time, our experiments demonstrated that Neural IR models require significantly longer indexing durations compared to BM25. On DAIC, indexing 1 million documents took approximately 65,000 seconds (18 hours) for ColBERT, 50,000 seconds (14 hours) for Hybrid, and 10,000 seconds (2.8 hours) for SBERT. Given that indexing 60 million documents on a single high-end GPU would take several weeks, Europeana would need a multi-GPU setup to reduce indexing time to a feasible range.

6

Conclusion

This research represents the first step in Europeana's exploration of neural IR and its potential to improve their search system by handling multilingual and heterogeneous metadata collections more effectively than the current BM25-based approach. Our study aimed to assess the feasibility, challenges, and benefits of adopting Neural IR models to enhance multilingual and heterogeneous metadata retrieval, providing a foundation for future developments in Europeana's search infrastructure.

6.1. Benefits and limits of neural IR

Based on this investigation and our quantitative/qualitative results, we cannot say that the performance of the neural models is absolutely better or worse than Europeanas current BM25 set up. Mainly because we do not have real judgements on the data, therefore, it is hard to state the absolute effectiveness of these models and give an absolute outcome. However, despite this limitation, we can still answer the main research question as our analysis highlights both advantages and challenges of integrating Neural IR into Europeana's search system:

Semantic retrieval performance
Based on our quantitative and qualitative analysis we can see that the neural IR models demonstrate useful semantic retrieval capabilities. ColBERT and Hybrid models performed decently even within a biased evaluation, suggesting their true effectiveness may be understated. Our qualitative analysis further showed that Neural IR models were able to more appropriately retrieve relevant documents

However, Neural IR models may struggle with single-entity queries, as they rely on contextual embeddings rather than exact term matches. This was evident in cases where BM25 outperformed Neural IR in retrieving precise entity-based results. A hybrid approach, where BM25 assists with entity matching while Neural IR enhances semantic retrieval, could mitigate this limitation.

Overall, Neural IR improves multilingual and concept-based retrieval but may need additional strategies for entity-focused searches to fully optimize Europeana's search.

Multilingualism
The quantitative analysis demonstrates that Neural IR models exhibit a more balanced and inherently multilingual language distribution, remaining largely unaffected by query and document augmentations. This suggests that Neural IR models are naturally better suited for cross-lingual retrieval compared to traditional lexical-based methods like BM25.

Our findings indicate that Neural IR models consistently retrieve documents in multiple languages, even without explicit translation or enrichment. Unlike BM25, which relies heavily on augmentations to improve multilingual retrieval, Neural models appear to capture semantic relationships across languages more effectively, making them more adaptable to Europeana's diverse metadata collection.

Saving costs on augmentations
The inherent multilingualism of the models means that Europeana can leverage Neural IR for multilingual retrieval without the additional costs associated with enrichment and translations.

However, when assessing the current value of these processes, our analysis indicates that enrichments and translations do help the current solr-BM25 method become more multilingual. And so, if Europeana continues using Solr-BM25, maintaining and expanding translation and enrichment efforts will be essential. In this case, investing in translating the remaining dataset, and expanding their pilot project of translating queries as well, could further improve retrieval quality for non-English queries and documents.

Adaptability with fine-tuning
Fine-tuning plays a crucial role in adapting Neural IR models to Europeana's domain, offering significant performance improvements and enhanced multilingual retrieval.

Our quantitative analysis shows that fine-tuning allows models to better align with Europeana's diverse metadata collection, optimizing search relevance while maintaining robust multilingual capabilities. For ColBERT, fine-tuning enhances retrieval effectiveness without disrupting language balance. Hybrid models also benefit from fine-tuning, particularly when combined with document augmentations, as they improve in multilingualism and performance. However, given the relatively small improvements observed, fine-tuning may not be strictly necessary for all configurations and should be weighed against computational costs and complexity.

Sensitivity of neural models
Our analysis indicates that Neural IR models, particularly fine-grained models like ColBERT, are highly sensitive to changes in document structure and contents. Due to their token-level matching approach, even minor modifications in formatting or content can lead to significant shifts in retrieval results. This sensitivity likely stems from how document embeddings are structured—small textual changes can substantially alter the multi-vector representation, affecting ranking behavior.

Additionally, fine-tuned models exhibit notable ranking variations, as seen in our qualitative analysis. In some cases, fine-tuning even results in worse performance, highlighting the importance of carefully evaluating training data and model updates. Given that Europeana's dataset is constantly evolving, this sensitivity must be considered, as changes to documents and re-training can influence embeddings and impact retrieval quality over time.

Neural IR in Europeana
Overall, to answer the main research question: Neural IR has the potential improve Europeana's search by enhancing multilingual retrieval, improving semantic search, and reducing reliance on augmentations like translations and enrichment's. Unlike BM25, which depends on exact term matching, Neural IR models capture semantic relationships across languages, enabling better cross-lingual retrieval without manual translations. Additionally, fine-tuning allows Neural IR models to adapt to Europeana's metadata, improving search relevance. While BM25 is better at single-entity queries, Neural IR offers more concept-based retrieval. However, its sensitivity to data changes requires careful infrastructure and evaluation planning for effective implementation.

6.2. Recommendation

Given our understanding of how Neural IR can be used to improve Europeanas search, we recommend that Europeana further explore and research the Hybrid BGE-M3 model that combines the strengths of lexical matching and Neural IR.

Given the limitations of the current evaluation framework, we cannot recommend a complete transition to Neural IR at this stage. Instead, Europeana should continue investigating the feasibility and effectiveness of a hybrid system, leveraging their current work on how to improve multilingualism in their current systems.

Our analysis outlines that the Hybrid system is an ideal candidate for future research and development as it has a lot of benefits of the neural approaches leveraging dense embeddings for semantic retrieval while managing to avoid some of the pitfalls of pure dense retrieval models.

Despite the pseudo-judgement-biased evaluation favoring BM25, the Hybrid model still performs well in quantitative experiments as they have moderately high recall values of around 0.6-0.7. Furthermore, our qualitative analysis affirms its effectiveness, showing that it retrieves relevant documents for queries where our biased quantitative analysis suggests otherwise.

Regarding the treatments, fine-tuning the Hybrid model does improve its multilinguality. The query augmentations do not have much of an impact on the retrieval of this model and are not necessary for this system. However the document augmentations do help improve the performance by assisting with the lexical matching.

The Hybrid model demonstrates greater stability and resilience, particularly in fine-tuning and handling noisy data, as shown in the qualitative analysis. Unlike purely dense models, it maintains performance consistency with the shifts in ranking caused by fine-tuning and changes in the document, making it more reliable for Europeana's evolving dataset. Additionally, by combining lexical and semantic signals, the Hybrid model is less sensitive to metadata inconsistencies and noisy text, which are common challenges in cultural heritage collections.

From a practical and implementation standpoint the Hybrid model also presents notable advantages over the other neural models. While ColBERT and Hybrid support long input sequences, we did not need to reduce the input length for Hybrid to fit within memory constraints, whereas ColBERT required reducing token limit from 8192 to 2048 tokens to run in our infrastructure. Additionally, the Hybrid model has a smaller index size and lower indexing time than ColBERT, making it a more feasible option for large-scale indexing—especially if chunking strategies are applied. Another benefit is the customization which milvus offers that ColBERT does not due to FAISS being in built into the Standford library.

Overall, the fine-tuned Hybrid BGE-M3 model with full document augmentations shows strong potential as an end-to-end retriever for improving multilingual retrieval and search performance in Europeana's search engine. Given its effectiveness, it warrants further exploration to assess its scalability and integration within Europeana's infrastructure.

7

Limitations and future work

7.1. Limitations

While our results provide valuable insights into multilingual retrieval, several implementation choices also have influenced our findings. This section examines key aspects of our evaluation setup, dataset construction, and implementation choices that could have shaped the observed trends, highlighting both potential limitations and areas for future refinement.

7.1.1. Dataset

judgements

The most significant limitation of our investigation is the absence of ground-truth relevance judgements, requiring us to rely on alternative approaches and additional analyses to compensate. While this does not invalidate our quantitative and qualitative findings, it does mean that our results are not conclusive. Instead, they should be interpreted as guiding insights rather than definitive answers. However, our analysis provides a strong foundation for future research, helping to refine evaluation strategies and inform the development of more robust neural IR models for Europeana.

Documents

The way we structured and formatted our dataset had a direct impact on retrieval performance and the conclusions we can draw from our results. Several key factors, including document structuring, language selection, and assumptions about enrichment and translation availability, likely influenced the behavior of the models.

One important design choice was the document formatting. We opted for a structured format that retained field names for the neural models. However, this may not have been the optimal decision, as structuring documents as free text without the field names could have led to a more uniform representation across models. The inclusion of field names may have added noise, particularly for SBERT, which lacks fine-grained token interactions as shown in the qualitative analysis. It is also unclear how retrieval performance would change if queries targeted specific fields. For example, a common field present in nearly all documents is "data Provider", and if a user were to query "provider", the neural system might prioritize this field rather than the contents of the field.

Language selection also played a significant role. Our dataset excluded low-represented languages such as Maltese and Greek, raising questions about whether their inclusion would have influenced multilingual retrieval effectiveness. Additionally, while some non-Latin scripts (e.g., Bulgarian Cyrillic) were included, we did not explicitly evaluate their applicability within the neural models. Just to clarify, these languages were not targeted, but part of documents with mixed languages (an effect of having code-switched data). A possible extension of this work would be to focus on non-Latin languages separately, rather than treating them within the same process as Latin-based languages.

Another key assumption in our dataset construction was that enrichments and translations were universally available in Europeana's data. However, in reality, not all documents are enriched or

translated in Europeanas actual production environment, meaning our evaluation does not fully reflect real-world retrieval conditions. Future investigations should consider how neural information retrieval models perform in settings where enrichment and translation coverage is inconsistent, rather than assuming a fully enriched and translated dataset.

Finally, the number of languages in our dataset may have influenced the training and inference of our neural models. Typical MLIR studies focus on fewer languages, while our dataset included a mix of 20 different languages. This raises questions about whether a singular retrieval process is equally effective for both high-resource and low-resource languages. A more refined approach might involve tailoring retrieval strategies separately for high- and low-resource languages, rather than applying the same process to all languages equally.

7.1.2. Model

When examining the implications of our implementation choices, we consider both the selection of models and the specific setup configurations used. These decisions, including how models were configured and and applied, had a significant influence on retrieval performance and overall effectiveness.

One key decision was not chunking documents for Hybrid and ColBERT models. Both models support long input sequences, with ColBERT allowing up to 8192 tokens, but this caused GPU memory exhaustion. To ensure feasibility, we lowered ColBERT's max_doclen to 2048 tokens, which fit within memory constraints. Hybrid, on the other hand, does not allow direct control over token limits but was able to process documents up to 8192 tokens. However, for very large documents, truncation may have occurred, potentially affecting retrieval effectiveness. The impact of this decision remains difficult to measure objectively due to the bias introduced by our pseudo-judgements, making it unclear whether chunking would have significantly improved results or simply changed ranking behavior.

Despite not chunking, we were still able to gain valuable insights into model behavior, particularly in how Neural IR models handle multilingual metadata and retrieval across different document structures. However, our findings were ultimately constrained by evaluation bias, which limits how conclusively we can determine the full extent of chunking's impact. Future work should explore chunking strategies alongside more reliable evaluation methods to better assess the trade-offs between full-document retrieval and passage-based approaches.

Additionally, our quantitative and qualitative analysis revealed that SBERT's fine-tuning was likely not done well, as its retrieval results were significantly worse than expected. Further investigation into the fine-tuning procedure and alternative training configurations is necessary to assess whether SBERT can perform better with the correct optimization.

7.1.3. Implementation issues

During our investigation, we encountered various implementation challenges that required us to develop multiple workarounds to ensure progress.

Our initial plan was to conduct the indexing, fine-tuning, and retrieval on Europeana's RND-3 server. However, we encountered significant resource limitations, making it impractical to run Neural IR models efficiently, ultimately costing us valuable time.

We faced serious issues with indexing the models on the server as it only had a 8 GB of GPU memory and would crash upon loading the models and indexes. Diagnosing this issue was particularly difficult, as the cause of the crashes was not immediately clear. Initial debugging suggested potential software conflicts or model errors, but further investigation revealed that memory limitations were the primary constraint.

This realization ultimately led us to transition to the Delft AI Cluster (DAIC), which provided significantly better computational resources, allowing us to properly run and evaluate our Neural IR models. This switch required modifications to our implementation.

On RND-3, we were able to use Docker to initialize and run Milvus-GPU, but DAIC's HPC environment did not support Docker, forcing us to use Milvus-Lite via Apptainer instead.

Overall, these infrastructure differences introduced additional complexities, requiring adjustments to

our workflow across different systems costing a lot of time.

7.2. Costs and future work

Implementing Neural IR at scale within Europeana will require significant investment. As outlined in our infrastructural analysis, scalable GPU resources, large storage capacity, and efficient indexing pipelines are necessary to maintain Neural IR performance over time. Additionally, Neural IR models require ongoing maintenance leading to higher operational costs compared to the current BM25 setup. While FAISS and Milvus offer fast querying, scaling Neural IR to tens of millions of documents will require careful optimization to ensure retrieval efficiency does not degrade as the collection grows.

Considering our primary recommendation of investigating a Hybrid approach, future work should primarily entail forming a complete evaluation framework for judgements. This would provide a more reliable measure of retrieval effectiveness and help assess the practical impact of Neural IR in Europeana. And exploring adaptive retrieval strategies that dynamically balance lexical and neural retrieval based on user intent. This could involve developing query classification techniques to determine when to rely more on BM25 for entity-based searches and when to leverage dense embeddings for semantic understanding, ultimately improving retrieval precision and efficiency.

Other aveneues for future work could focus on expanding fine-tuning experiments, and assessing alternative architectures. Another promising direction is using Neural IR as a reranker, where a lexical retriever (BM25) retrieves initial candidates, and a fine-tuned Hybrid model refines the rankings. Since Europeanas collection is also always expanding and changing, looking into re-indexing strategies in Milvus should also be considered. Additionally, exploring semantic chunking strategies could enhance indexing efficiency while maintaining retrieval quality. Finally, future work can investigate how low-resource languages, such as Maltese, can be better handled, ensuring that Europeana's search remains inclusive and effective for all users.

References

[1] Jianlv Chen et al. "BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation". In: (Feb. 2024). URL: http://arxiv.org/abs/2402.03216.

[2] Rohan Jha et al. "Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever". In: (Aug. 2024). URL: http://arxiv.org/abs/2408.16672.

[3] Suraj Nair et al. *Learning a Sparse Representation Model for Neural CLIR*. 2022. URL: http://ceur-ws.org.

[4] Vladimir Karpukhin et al. *Dense Passage Retrieval for Open-Domain Question Answering*. URL: https://github.com/facebookresearch/DPR..

[5] Joel Azzopardi. "Translating Justice: A Cross-Lingual Information Retrieval System for Maltese Case Law Documents". In: vol. 14612 LNCS. Springer Science and Business Media Deutschland GmbH, 2024, pp. 236–240. ISBN: 9783031560682. DOI: 10.1007/978-3-031-56069-9_24.

[6] Joel Barnard. *What is embedding?* Dec. 2024. URL: https://www.ibm.com/think/topics/embedding?utm_source=chatgpt.com.

[7] Luiz Bonifacio et al. *mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset*. 2022. arXiv: 2108.13897 [cs.CL]. URL: https://arxiv.org/abs/2108.13897.

[8] Leonid Boytsov et al. "Understanding Performance of Long-Document Ranking Models through Comprehensive Evaluation and Leaderboarding". In: (July 2022). URL: http://arxiv.org/abs/2207.01262.

[9] A. Chayapathi et al. "Usage of Multilingual Indexing for Retrieving the Information in Multiple Language". In: Jan. 2021, pp. 255–264. ISBN: 978-981-15-5242-7. DOI: 10.1007/978-981-15-5243-4_22.

[10] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: (Nov. 2019). URL: http://arxiv.org/abs/1911.02116.

[11] Matteo Corsi and Julián Urbano. "The Treatment of Ties in Rank-Biased Overlap". In: Association for Computing Machinery, Inc, July 2024, pp. 251–260. ISBN: 9798400704314. DOI: 10.1145/3626772.3657700.

[12] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *none* (Oct. 2018). URL: http://arxiv.org/abs/1810.04805.

[13] Atsushi Fujii and Tetsuya Ishikawa. *Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration*. 2002. arXiv: cs/0206015 [cs.CL]. URL: https://arxiv.org/abs/cs/0206015.

[14] Sakib Haque et al. "Semantic Similarity Metrics for Evaluating Source Code Summarization". In: vol. 2022-March. IEEE Computer Society, 2022, pp. 36–47. ISBN: 9781450392983. DOI: 10.1145/nnnnnnn.nnnnnnn.

[15] Omar Khattab and Matei Zaharia. "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT". In: Association for Computing Machinery, Inc, July 2020, pp. 39–48. ISBN: 9781450380164. DOI: 10.1145/3397271.3401075.

[16] Marijn Koolen et al. "A cross-language approach to historic document retrieval". In: vol. 3936 LNCS. 2006, pp. 407–419. ISBN: 3540333479. DOI: 10.1007/11735106_36.

[17] Guillaume Lample and Alexis Conneau. "Cross-lingual Language Model Pretraining". In: (Jan. 2019). URL: http://arxiv.org/abs/1901.07291.

[18] Dawn Lawrie et al. "Neural Approaches to Multilingual Information Retrieval". In: (Sept. 2022). URL: http://arxiv.org/abs/2209.01335.

[19] Shihao Liang et al. “Exploring Format Consistency for Instruction Tuning”. In: (July 2023). URL: http://arxiv.org/abs/2307.15504.

[20] Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. “How Language-Neutral is Multilingual BERT?” In: *none* (Nov. 2019). URL: http://arxiv.org/abs/1911.03310.

[21] Robert Litschko, Ekaterina Artemova, and Barbara Plank. “Boosting Zero-shot Cross-lingual Retrieval by Training on Artificially Code-Switched Data”. In: (May 2023). URL: http://arxiv.org/abs/2305.05295.

[22] Mónica Marrero and Antoine Isaac. “Implementation and Evaluation of a Multilingual Search Pilot in the Europeana Digital Library”. In: vol. 13541 LNCS. Springer Science and Business Media Deutschland GmbH, 2022, pp. 93–106. ISBN: 9783031168017. DOI: 10.1007/978-3-031-16802-4_8.

[23] Bhaskar Mitra and Nick Craswell. “An Introduction to Neural Information Retrieval”. In: *Foundations and Trends® in Information Retrieval* 13.1 (2018), pp. 1–126. ISSN: 1554-0669. DOI: 10.1561/1500000061. URL: http://dx.doi.org/10.1561/1500000061.

[24] Dan Munteanu. *VECTOR SPACE MODEL FOR DOCUMENT REPRESENTATION IN INFORMATION RETRIEVAL*. 2007.

[25] Thanh-Do Nguyen et al. *Passage-based BM25 Hard Negatives: A Simple and Effective Negative Sampling Strategy For Dense Retrieval*.

[26] Tord Nilsen. “Semantic search in an online collection”. In: (2023). URL: https://beta.nasjonalmuseet.no/2023/08/add-semantic-search-to-a-online-collection/.

[27] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. *Word2Sense : Sparse Interpretable Word Embeddings*.

[28] PothulaSujatha and Dhavachelvan. “A Review on the Cross and Multilingual Information Retrieval”. In: *International journal of Web Semantic Technology* 2 (4 Oct. 2011), pp. 115–124. ISSN: 09762280. DOI: 10.5121/ijwest.2011.2409.

[29] Thilina Chaturanga Rajapakse, Andrew Yates, and Maarten De Rijke. “Negative Sampling Techniques for Dense Passage Retrieval in a Multilingual Setting”. In: Association for Computing Machinery, Inc, July 2024, pp. 575–584. ISBN: 9798400704314. DOI: 10.1145/3626772.3657854.

[30] Prashanth Rao. *Vector databases (2): Understanding their internals — thedataquarry.com*. https://thedataquarry.com/posts/vector-db-2/. [Accessed 16-12-2024].

[31] Nils Reimers and Iryna Gurevych. *Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation*. 2020. arXiv: 2004.09813 [cs.CL]. URL: https://arxiv.org/abs/2004.09813.

[32] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: (Aug. 2019). URL: http://arxiv.org/abs/1908.10084.

[33] Stephen Robertson. *Understanding Inverse Document Frequency: On theoretical arguments for IDF*.

[34] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Foundations and Trends in Information Retrieval* 3 (Jan. 2009), pp. 333–389. DOI: 10.1561/1500000019.

[35] Stephen Robertson and Hugo Zaragoza. “The probabilistic relevance framework: BM25 and beyond”. In: *Foundations and Trends in Information Retrieval* 3 (4 2009), pp. 333–389. ISSN: 15540669. DOI: 10.1561/1500000019.

[36] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *none* (Oct. 2019). URL: http://arxiv.org/abs/1910.01108.

[37] Rajendra P Srivastava. *A New Measure of Similarity in Textual Analysis: Vector Similarity Metric versus Cosine Similarity Metric*. URL: https://www.researchgate.net/publication/357159649.

[38] Nicola Tonellotto. “Lecture Notes on Neural Information Retrieval”. In: *none* (July 2022). URL: http://arxiv.org/abs/2207.13443.

[39] “Translate-Distill: Learning Cross-Language Dense Retrieval by Translation and Distillation”. In: (Jan. 2024). URL: http://arxiv.org/abs/2401.04810.

[40] *Trec &x2014; ir-kit documentation — ir-kit.readthedocs.io*. https://ir-kit.readthedocs.io/en/latest/trec.html. [Accessed 17-12-2024].

[41] Jiapeng Wang and Yihong Dong. "Measurement of Text Similarity: A Survey". In: *Information* 11 (Aug. 2020), p. 421. DOI: 10.3390/info11090421.

[42] Tong Wang, Ninad Kulkarni, and Yanjun Qi. "Less is More for Improving Automatic Evaluation of Factual Consistency". In: (Apr. 2024). URL: http://arxiv.org/abs/2404.06579.

[43] Jitao Xu and François Yvon. "Can You Traducir This? Machine Translation for Code-Switched Input". In: (May 2021). URL: http://arxiv.org/abs/2105.04846.

[44] Eugene Yang, Dawn Lawrie, and James Mayfield. "Distillation for Multilingual Information Retrieval". In: Association for Computing Machinery, Inc, July 2024, pp. 2368–2373. ISBN: 9798400704314. DOI: 10.1145/3626772.3657955.

[45] Yutao Zhu et al. "Large Language Models for Information Retrieval: A Survey". In: (Aug. 2023). URL: http://arxiv.org/abs/2308.07107.

A

Initial quantitative results

This section presents all of the results from our quantitative analysis

A.0.1. Absolute results

System	N_{same}	N_{EN}	N_{diff}	pct_{same}	pct_{EN}	pct_{diff}	H_{same}	H_{EN}	H_{diff}
B-O-P	70131.00	68916.00	14067.00	45.80	45.01	9.19	2.46	0.67	6.16
B-O-PE	70403.00	74094.00	17041.00	43.58	45.87	10.55	2.45	0.70	6.29
B-O-PT	60528.00	108082.00	14781.00	33.00	58.94	8.06	2.72	0.84	6.22
B-O-PET	61291.00	110807.00	17707.00	32.29	58.38	9.33	2.70	0.88	6.31
B-OT-P	67394.00	86443.00	27219.00	37.22	47.74	15.03	2.39	2.33	6.71
B-OT-PE	68164.00	91530.00	29078.00	36.11	48.49	15.40	2.39	2.33	6.72
B-OT-PT	58989.00	118741.00	49604.00	25.95	52.23	21.82	2.66	2.23	6.78
B-OT-PET	60310.00	121608.00	50362.00	25.96	52.35	21.68	2.65	2.23	6.78
CZ-O-P	230634.00	463284.00	217282.00	25.31	50.84	23.85	2.67	1.79	6.86
CZ-O-PE	235488.00	461655.00	214057.00	25.84	50.66	23.49	2.70	1.70	6.84
CZ-O-PT	223877.00	468196.00	219127.00	24.57	51.38	24.05	2.72	1.79	6.86
CZ-O-PET	232548.00	465519.00	213133.00	25.52	51.09	23.39	2.74	1.70	6.84
CZ-OT-P	222067.00	467591.00	221542.00	24.37	51.32	24.31	2.63	1.91	6.87
CZ-OT-PE	225559.00	467003.00	218638.00	24.75	51.25	23.99	2.65	1.87	6.85
CZ-OT-PT	215802.00	470892.00	224506.00	23.68	51.68	24.64	2.68	1.89	6.87
CZ-OT-PET	223111.00	469643.00	218446.00	24.49	51.54	23.97	2.70	1.85	6.84
CF-O-P	242975.00	463918.00	204307.00	26.67	50.91	22.42	2.77	1.57	6.77
CF-O-PE	248822.00	461097.00	201281.00	27.31	50.60	22.09	2.75	1.61	6.74
CF-O-PT	246333.00	463239.00	201628.00	27.03	50.84	22.13	2.76	1.63	6.73
CF-O-PET	248360.00	460883.00	201957.00	27.26	50.58	22.16	2.73	1.62	6.73
CF-OT-P	231291.00	469637.00	210272.00	25.38	51.54	23.08	2.72	1.81	6.76
CF-OT-PE	237267.00	467096.00	206837.00	26.04	51.26	22.70	2.70	1.83	6.75
CF-OT-PT	234672.00	468242.00	208286.00	25.75	51.39	22.86	2.71	1.82	6.74
CF-OT-PET	237080.00	466627.00	207493.00	26.02	51.21	22.77	2.68	1.82	6.73

Table A.1: Absolute results for all BM25 and Colbert systems: language distribution

System	N_{same}	N_{EN}	N_{diff}	pct_{same}	pct_{EN}	pct_{diff}	H_{same}	H_{EN}	H_{diff}
SZ-O-P	246235.00	456429.00	208536.00	27.02	50.09	22.89	1.69	2.29	6.23
SZ-O-PE	250197.00	455666.00	205337.00	27.46	50.01	22.53	1.66	2.30	6.21
SZ-O-PT	234825.00	462230.00	214145.00	25.77	50.73	23.50	1.76	2.30	6.23
SZ-O-PET	247343.00	457067.00	206790.00	27.14	50.16	22.69	1.68	2.30	6.22
SZ-OT-P	251210.00	452212.00	207778.00	27.57	49.63	22.80	1.72	2.25	6.20
SZ-OT-PE	254903.00	451561.00	204736.00	27.97	49.56	22.47	1.69	2.26	6.17
SZ-OT-PT	239940.00	457657.00	213603.00	26.33	50.23	23.44	1.78	2.25	6.20
SZ-OT-PET	252108.00	452647.00	206445.00	27.67	49.68	22.66	1.71	2.26	6.19
SF-O-P	188236.00	468049.00	254915.00	20.66	51.37	27.98	1.49	2.38	6.38
SF-O-PE	16921.00	512119.00	382141.00	1.86	56.20	41.94	1.52	2.12	3.54
SF-O-PT	185851.00	473305.00	252044.00	20.40	51.94	27.66	1.40	2.41	6.46
SF-O-PET	166686.00	477967.00	266547.00	18.29	52.45	29.25	1.50	2.38	6.70
SF-OT-P	189916.00	463181.00	258103.00	20.84	50.83	28.33	1.53	2.32	6.42
SF-OT-PE	16986.00	513491.00	380704.00	1.86	56.35	41.78	1.56	2.35	3.60
SF-OT-PT	187368.00	465758.00	258074.00	20.56	51.11	28.32	1.43	2.34	6.51
SF-OT-PET	167622.00	473127.00	270451.00	18.40	51.92	29.68	1.52	2.33	6.73
HZ-O-P	357043.00	393714.00	160442.00	39.18	43.21	17.61	2.24	1.83	6.70
HZ-O-PE	364799.00	387912.00	158487.00	40.04	42.57	17.39	2.19	1.82	6.71
HZ-O-PT	345540.00	405103.00	160556.00	37.92	44.46	17.62	2.29	1.88	6.72
HZ-O-PET	356706.00	394857.00	159635.00	39.15	43.33	17.52	2.22	1.85	6.71
HZ-OT-P	336531.00	414181.00	160487.00	36.93	45.45	17.61	2.11	2.07	6.68
HZ-OT-PE	344582.00	409503.00	157113.00	37.82	44.94	17.24	2.07	2.06	6.69
HZ-OT-PT	326628.00	420001.00	164570.00	35.85	46.09	18.06	2.17	2.07	6.71
HZ-OT-PET	337585.00	413154.00	160459.00	37.05	45.34	17.61	2.11	2.06	6.69
HF-O-P	377678.00	389215.00	144307.00	41.45	42.71	15.84	2.24	1.84	6.55
HF-O-PE	362338.00	390768.00	158094.00	39.76	42.88	17.35	2.18	1.87	6.75
HF-O-PT	320163.00	431827.00	159210.00	35.14	47.39	17.47	2.53	1.91	6.57
HF-O-PET	316116.00	427825.00	167258.00	34.69	46.95	18.36	2.50	1.84	6.57
HF-OT-P	352210.00	410439.00	148551.00	38.65	45.04	16.30	2.11	2.08	6.47
HF-OT-PE	340869.00	410694.00	159637.00	37.41	45.07	17.52	2.04	2.08	6.68
HF-OT-PT	296296.00	441099.00	173805.00	32.52	48.41	19.07	2.42	2.05	6.51
HF-OT-PET	293803.00	439076.00	178320.00	32.24	48.19	19.57	2.39	2.03	6.50

Table A.2: Absolute results for all SBERT and Hybrid systems: language distribution

System	AP_{EN}	AP_{NEN}	R_{EN}	$R_N EN$	RR_{EN}	RR_{NEN}
B-O-P	0.62	0.75	0.76	0.89	0.68	0.79
B-O-PE	0.63	0.76	0.77	0.90	0.69	0.80
B-O-PT	0.72	0.76	0.90	0.90	0.77	0.80
B-O-PET	0.73	0.78	0.91	0.91	0.78	0.81
B-OT-P	0.62	0.69	0.76	0.88	0.68	0.73
B-OT-PE	0.63	0.69	0.77	0.89	0.69	0.73
B-OT-PT	0.72	0.69	0.90	0.89	0.77	0.73
B-OT-PET	0.73	0.70	0.91	0.90	0.78	0.74
CZ-O-P	0.22	0.32	0.52	0.62	0.25	0.36
CZ-O-PE	0.23	0.34	0.54	0.64	0.27	0.38
CZ-O-PT	0.21	0.31	0.54	0.62	0.25	0.35
CZ-O-PET	0.23	0.34	0.56	0.65	0.27	0.38
CZ-OT-P	0.22	0.29	0.52	0.61	0.25	0.32
CZ-OT-PE	0.23	0.30	0.54	0.63	0.27	0.34
CZ-OT-PT	0.21	0.28	0.54	0.62	0.25	0.32
CZ-OT-PET	0.23	0.31	0.56	0.64	0.27	0.34
CF-O-P	0.27	0.38	0.57	0.64	0.32	0.43
CF-O-PE	0.26	0.37	0.55	0.63	0.30	0.42
CF-O-PT	0.27	0.37	0.59	0.65	0.31	0.42
CF-O-PET	0.25	0.36	0.55	0.63	0.30	0.41
CF-OT-P	0.27	0.34	0.57	0.65	0.32	0.39
CF-OT-PE	0.26	0.33	0.55	0.63	0.30	0.37
CF-OT-PT	0.27	0.33	0.59	0.65	0.31	0.38
CF-OT-PET	0.25	0.32	0.55	0.63	0.30	0.37
SZ-O-P	0.05	0.03	0.16	0.14	0.06	0.04
SZ-O-PE	0.04	0.03	0.15	0.13	0.06	0.04
SZ-O-PT	0.05	0.03	0.17	0.12	0.07	0.03
SZ-O-PET	0.05	0.03	0.16	0.14	0.06	0.04
SZ-OT-P	0.05	0.04	0.16	0.16	0.06	0.05
SZ-OT-PE	0.04	0.04	0.15	0.15	0.06	0.05
SZ-OT-PT	0.05	0.03	0.17	0.14	0.07	0.04
SZ-OT-PET	0.05	0.04	0.16	0.15	0.06	0.05
SF-O-P	0.00	0.00	0.02	0.01	0.01	0.00
SF-O-PE	0.00	0.00	0.00	0.00	0.00	0.00
SF-O-PT	0.00	0.00	0.03	0.01	0.01	0.00
SF-O-PET	0.00	0.00	0.02	0.01	0.01	0.00
SF-OT-P	0.00	0.00	0.02	0.01	0.01	0.00
SF-OT-PE	0.00	0.00	0.00	0.00	0.00	0.00
SF-OT-PT	0.00	0.00	0.03	0.02	0.01	0.00
SF-OT-PET	0.00	0.00	0.02	0.01	0.01	0.00
HZ-O-P	0.27	0.38	0.61	0.70	0.31	0.43
HZ-O-PE	0.26	0.37	0.59	0.69	0.31	0.42
HZ-O-PT	0.28	0.37	0.64	0.70	0.33	0.42
HZ-O-PET	0.27	0.36	0.61	0.69	0.32	0.41
HZ-OT-P	0.27	0.31	0.61	0.69	0.31	0.36
HZ-OT-PE	0.26	0.31	0.59	0.69	0.31	0.35
HZ-OT-PT	0.28	0.33	0.64	0.70	0.33	0.37
HZ-OT-PET	0.27	0.32	0.61	0.69	0.32	0.36
HF-O-P	0.25	0.34	0.54	0.62	0.30	0.39
HF-O-PE	0.22	0.27	0.48	0.58	0.27	0.32
HF-O-PT	0.29	0.35	0.62	0.63	0.34	0.40
HF-O-PET	0.28	0.36	0.61	0.65	0.34	0.41
HF-OT-P	0.25	0.27	0.54	0.62	0.30	0.31
HF-OT-PE	0.22	0.21	0.48	0.58	0.27	0.24
HF-OT-PT	0.29	0.30	0.62	0.64	0.34	0.34
HF-OT-PET	0.28	0.30	0.61	0.65	0.34	0.34

Table A.3: Absolute results for all systems: performance metrics

A.0.2. Comparative results

Model comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	CZ-O-P	-0.41	-0.43	-0.24	-0.27	-0.43	-0.43	0.24	0.34
B-O-PE	CZ-O-PE	-0.40	-0.42	-0.23	-0.26	-0.42	-0.41	0.25	0.35
B-O-PT	CZ-O-PT	-0.51	-0.45	-0.36	-0.28	-0.52	-0.45	0.23	0.33
B-O-PET	CZ-O-PET	-0.50	-0.43	-0.35	-0.26	-0.51	-0.43	0.25	0.35
B-OT-P	CZ-OT-P	-0.41	-0.40	-0.24	-0.27	-0.43	-0.41	0.24	0.30
B-OT-PE	CZ-OT-PE	-0.40	-0.39	-0.23	-0.26	-0.42	-0.39	0.25	0.32
B-OT-PT	CZ-OT-PT	-0.51	-0.41	-0.36	-0.27	-0.52	-0.42	0.23	0.29
B-OT-PET	CZ-OT-PET	-0.50	-0.40	-0.35	-0.26	-0.51	-0.40	0.25	0.31
B-O-P	SZ-O-P	-0.57	-0.72	-0.60	-0.75	-0.62	-0.75	0.05	0.03
B-O-PE	SZ-O-PE	-0.58	-0.73	-0.62	-0.76	-0.63	-0.76	0.05	0.03
B-O-PT	SZ-O-PT	-0.67	-0.74	-0.72	-0.78	-0.70	-0.77	0.05	0.02
B-O-PET	SZ-O-PET	-0.69	-0.74	-0.75	-0.78	-0.71	-0.77	0.05	0.03
B-OT-P	SZ-OT-P	-0.57	-0.65	-0.60	-0.72	-0.62	-0.67	0.05	0.05
B-OT-PE	SZ-OT-PE	-0.58	-0.65	-0.62	-0.74	-0.63	-0.68	0.05	0.04
B-OT-PT	SZ-OT-PT	-0.67	-0.66	-0.72	-0.74	-0.70	-0.69	0.05	0.03
B-OT-PET	SZ-OT-PET	-0.69	-0.66	-0.75	-0.75	-0.71	-0.69	0.05	0.04
B-O-P	HZ-O-P	-0.36	-0.37	-0.16	-0.19	-0.37	-0.37	0.32	0.39
B-O-PE	HZ-O-PE	-0.37	-0.39	-0.18	-0.21	-0.38	-0.38	0.31	0.37
B-O-PT	HZ-O-PT	-0.44	-0.39	-0.25	-0.20	-0.44	-0.38	0.31	0.38
B-O-PET	HZ-O-PET	-0.46	-0.41	-0.29	-0.22	-0.46	-0.40	0.29	0.36
B-OT-P	HZ-OT-P	-0.36	-0.38	-0.16	-0.18	-0.37	-0.37	0.32	0.33
B-OT-PE	HZ-OT-PE	-0.37	-0.39	-0.18	-0.20	-0.38	-0.38	0.31	0.32
B-OT-PT	HZ-OT-PT	-0.44	-0.37	-0.25	-0.19	-0.44	-0.36	0.31	0.33
B-OT-PET	HZ-OT-PET	-0.46	-0.39	-0.29	-0.21	-0.46	-0.38	0.29	0.32

Table A.4: Performance metrics across different models and augmentation strategies

Finetuning comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
CZ-O-P	CF-O-P	0.05	0.06	0.05	0.02	0.06	0.07	0.25	0.29
CZ-O-PE	CF-O-PE	0.03	0.03	0.01	-0.00	0.04	0.04	0.26	0.31
CZ-O-PT	CF-O-PT	0.05	0.06	0.05	0.03	0.06	0.07	0.25	0.28
CZ-O-PET	CF-O-PET	0.02	0.02	-0.00	-0.02	0.03	0.03	0.26	0.31
CZ-OT-P	CF-OT-P	0.05	0.06	0.05	0.03	0.06	0.07	0.25	0.28
CZ-OT-PE	CF-OT-PE	0.03	0.03	0.01	0.00	0.04	0.04	0.26	0.30
CZ-OT-PT	CF-OT-PT	0.05	0.05	0.05	0.04	0.06	0.06	0.25	0.27
CZ-OT-PET	CF-OT-PET	0.02	0.02	-0.00	-0.01	0.03	0.02	0.26	0.29
SZ-O-P	SF-O-P	-0.04	-0.03	-0.14	-0.13	-0.06	-0.04	0.01	0.00
SZ-O-PE	SF-O-PE	-0.04	-0.03	-0.15	-0.13	-0.06	-0.04	0.00	0.00
SZ-O-PT	SF-O-PT	-0.05	-0.02	-0.15	-0.11	-0.06	-0.03	0.01	0.01
SZ-O-PET	SF-O-PET	-0.04	-0.03	-0.14	-0.13	-0.06	-0.04	0.01	0.00
SZ-OT-P	SF-OT-P	-0.04	-0.04	-0.14	-0.14	-0.06	-0.05	0.01	0.01
SZ-OT-PE	SF-OT-PE	-0.04	-0.04	-0.15	-0.15	-0.06	-0.05	0.00	0.00
SZ-OT-PT	SF-OT-PT	-0.05	-0.03	-0.15	-0.13	-0.06	-0.04	0.01	0.01
SZ-OT-PET	SF-OT-PET	-0.04	-0.04	-0.14	-0.14	-0.06	-0.05	0.01	0.00
HZ-O-P	HF-O-P	-0.01	-0.04	-0.07	-0.07	-0.01	-0.04	0.24	0.25
HZ-O-PE	HF-O-PE	-0.04	-0.10	-0.11	-0.11	-0.03	-0.10	0.19	0.19
HZ-O-PT	HF-O-PT	0.00	-0.02	-0.03	-0.07	0.01	-0.02	0.24	0.25
HZ-O-PET	HF-O-PET	0.01	-0.01	-0.01	-0.04	0.02	-0.01	0.24	0.26
HZ-OT-P	HF-OT-P	-0.01	-0.05	-0.07	-0.07	-0.01	-0.05	0.24	0.23
HZ-OT-PE	HF-OT-PE	-0.04	-0.10	-0.11	-0.10	-0.03	-0.10	0.19	0.17
HZ-OT-PT	HF-OT-PT	0.00	-0.03	-0.03	-0.06	0.01	-0.03	0.24	0.24
HZ-OT-PET	HF-OT-PET	0.01	-0.02	-0.01	-0.04	0.02	-0.02	0.24	0.24

Table A.5: Performance metrics across different models and augmentation strategies

Query augmentation comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	B-OT-P	0.00	-0.06	0.00	-0.01	0.00	-0.07	1.00	0.87
B-O-PE	B-OT-PE	0.00	-0.06	0.00	-0.01	0.00	-0.07	1.00	0.87
B-O-PT	B-OT-PT	0.00	-0.07	0.00	-0.01	0.00	-0.07	1.00	0.84
B-O-PET	B-OT-PET	0.00	-0.07	0.00	-0.01	0.00	-0.07	1.00	0.85
CZ-O-P	CZ-OT-P	0.00	-0.04	0.00	-0.01	0.00	-0.04	1.00	0.70
CZ-O-PE	CZ-OT-PE	0.00	-0.04	0.00	-0.01	0.00	-0.05	1.00	0.71
CZ-O-PT	CZ-OT-PT	0.00	-0.03	0.00	-0.00	0.00	-0.04	1.00	0.70
CZ-O-PET	CZ-OT-PET	0.00	-0.04	0.00	-0.01	0.00	-0.04	1.00	0.71
CF-O-P	CF-OT-P	0.00	-0.04	0.00	0.00	0.00	-0.04	1.00	0.71
CF-O-PE	CF-OT-PE	0.00	-0.04	0.00	-0.00	0.00	-0.05	1.00	0.71
CF-O-PT	CF-OT-PT	0.00	-0.04	0.00	0.00	0.00	-0.05	1.00	0.71
CF-O-PET	CF-OT-PET	0.00	-0.04	0.00	0.00	0.00	-0.05	1.00	0.71
SZ-O-P	SZ-OT-P	0.00	0.01	0.00	0.02	0.00	0.01	1.00	0.72
SZ-O-PE	SZ-OT-PE	0.00	0.01	0.00	0.01	0.00	0.01	1.00	0.72
SZ-O-PT	SZ-OT-PT	0.00	0.00	0.00	0.02	0.00	0.00	1.00	0.73
SZ-O-PET	SZ-OT-PET	0.00	0.01	0.00	0.01	0.00	0.01	1.00	0.73
SF-O-P	SF-OT-P	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.72
SF-O-PE	SF-OT-PE	0.00	-0.00	0.00	0.00	0.00	-0.00	1.00	0.93
SF-O-PT	SF-OT-PT	0.00	-0.00	0.00	0.00	0.00	-0.00	1.00	0.71
SF-O-PET	SF-OT-PET	0.00	-0.00	0.00	0.00	0.00	-0.00	1.00	0.70
HZ-O-P	HZ-OT-P	0.00	-0.06	0.00	-0.00	0.00	-0.07	1.00	0.63
HZ-O-PE	HZ-OT-PE	0.00	-0.06	0.00	-0.00	0.00	-0.07	1.00	0.63
HZ-O-PT	HZ-OT-PT	0.00	-0.05	0.00	0.00	0.00	-0.05	1.00	0.64
HZ-O-PET	HZ-OT-PET	0.00	-0.05	0.00	-0.00	0.00	-0.05	1.00	0.63
HF-O-P	HF-OT-P	0.00	-0.07	0.00	-0.00	0.00	-0.08	1.00	0.62
HF-O-PE	HF-OT-PE	0.00	-0.07	0.00	0.01	0.00	-0.08	1.00	0.61
HF-O-PT	HF-OT-PT	0.00	-0.06	0.00	0.01	0.00	-0.06	1.00	0.63
HF-O-PET	HF-OT-PET	0.00	-0.06	0.00	0.00	0.00	-0.06	1.00	0.63

Table A.6: Performance metrics across different models and augmentation strategies

Document enrichment comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	B-O-PE	0.00	0.01	0.01	0.01	0.00	0.00	0.95	0.97
B-O-PT	B-O-PET	0.01	0.01	0.01	0.01	0.01	0.01	0.96	0.97
B-OT-P	B-OT-PE	0.00	0.00	0.01	0.01	0.00	0.00	0.95	0.96
B-OT-PT	B-OT-PET	0.01	0.01	0.01	0.01	0.01	0.01	0.96	0.96
CZ-O-P	CZ-O-PE	0.01	0.02	0.02	0.01	0.01	0.02	0.32	0.37
CZ-O-PT	CZ-O-PET	0.02	0.03	0.02	0.03	0.02	0.03	0.31	0.36
CZ-OT-P	CZ-OT-PE	0.01	0.02	0.02	0.01	0.01	0.02	0.32	0.35
CZ-OT-PT	CZ-OT-PET	0.02	0.03	0.02	0.03	0.02	0.02	0.31	0.34
CF-O-P	CF-O-PE	-0.01	-0.01	-0.02	-0.01	-0.01	-0.01	0.31	0.37
CF-O-PT	CF-O-PET	-0.01	-0.01	-0.03	-0.02	-0.01	-0.01	0.31	0.36
CF-OT-P	CF-OT-PE	-0.01	-0.01	-0.02	-0.02	-0.01	-0.01	0.31	0.35
CF-OT-PT	CF-OT-PET	-0.01	-0.01	-0.03	-0.02	-0.01	-0.01	0.31	0.34
SZ-O-P	SZ-O-PE	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.82	0.82
SZ-O-PT	SZ-O-PET	-0.01	0.01	-0.02	0.01	-0.01	0.00	0.78	0.76
SZ-OT-P	SZ-OT-PE	-0.00	-0.01	-0.01	-0.01	-0.00	-0.00	0.82	0.80
SZ-OT-PT	SZ-OT-PET	-0.01	0.01	-0.02	0.00	-0.01	0.01	0.78	0.75
SF-O-P	SF-O-PE	-0.00	-0.00	-0.02	-0.01	-0.00	-0.00	0.00	0.00
SF-O-PT	SF-O-PET	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.03	0.02
SF-OT-P	SF-OT-PE	-0.00	-0.00	-0.02	-0.01	-0.00	-0.00	0.00	0.00
SF-OT-PT	SF-OT-PET	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.03	0.02
HZ-O-P	HZ-O-PE	-0.01	-0.01	-0.02	-0.01	-0.01	-0.01	0.74	0.72
HZ-O-PT	HZ-O-PET	-0.01	-0.01	-0.03	-0.00	-0.02	-0.01	0.73	0.72
HZ-OT-P	HZ-OT-PE	-0.01	-0.01	-0.02	-0.01	-0.01	-0.01	0.74	0.71
HZ-OT-PT	HZ-OT-PET	-0.01	-0.01	-0.03	-0.01	-0.02	-0.01	0.73	0.70
HF-O-P	HF-O-PE	-0.03	-0.07	-0.06	-0.05	-0.03	-0.07	0.35	0.35
HF-O-PT	HF-O-PET	-0.00	0.00	-0.01	0.02	-0.00	0.00	0.43	0.46
HF-OT-P	HF-OT-PE	-0.03	-0.06	-0.06	-0.04	-0.03	-0.07	0.35	0.31
HF-OT-PT	HF-OT-PET	-0.00	-0.00	-0.01	0.02	-0.00	0.00	0.43	0.42
B-O-P	B-O-PET	0.11	0.02	0.14	0.02	0.09	0.02	0.81	0.95
B-OT-P	B-OT-PET	0.11	0.01	0.14	0.02	0.09	0.01	0.81	0.86
CZ-O-P	CZ-O-PET	0.02	0.02	0.04	0.03	0.02	0.02	0.30	0.35
CZ-OT-P	CZ-OT-PET	0.02	0.02	0.04	0.03	0.02	0.02	0.30	0.33
CF-O-P	CF-O-PET	-0.01	-0.02	-0.02	-0.01	-0.02	-0.02	0.29	0.35
CF-OT-P	CF-OT-PET	-0.01	-0.02	-0.02	-0.02	-0.02	-0.02	0.29	0.34
SZ-O-P	SZ-O-PET	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.78	0.78
SZ-OT-P	SZ-OT-PET	-0.00	-0.01	-0.01	-0.01	-0.00	-0.00	0.78	0.76
SF-O-P	SF-O-PET	-0.00	-0.00	-0.01	-0.00	-0.00	0.00	0.02	0.01
SF-OT-P	SF-OT-PET	-0.00	-0.00	-0.01	-0.00	-0.00	0.00	0.02	0.01
HZ-O-P	HZ-O-PET	0.00	-0.01	0.01	-0.00	0.00	-0.01	0.69	0.68
HZ-OT-P	HZ-OT-PET	0.00	0.00	0.01	-0.00	0.00	0.00	0.69	0.66
HF-O-P	HF-O-PET	0.03	0.02	0.07	0.02	0.04	0.02	0.41	0.46
HF-OT-P	HF-OT-PET	0.03	0.03	0.07	0.03	0.04	0.03	0.41	0.39

Table A.7: Performance metrics across different models and augmentation strategies

Document translation comparison results

system$_1$	system$_2$	dAP$_{EN}$	dAP$_{NEN}$	dR$_{EN}$	dR$_N EN$	dRR$_{EN}$	dRR$_{NEN}$	RBO$_{EN}$	RBO$_{NEN}$
B-O-P	B-O-PT	0.10	0.01	0.13	0.01	0.08	0.01	0.82	0.96
B-O-PE	B-O-PET	0.11	0.02	0.13	0.01	0.09	0.02	0.83	0.97
B-OT-P	B-OT-PT	0.10	0.00	0.13	0.01	0.08	0.01	0.82	0.87
B-OT-PE	B-OT-PET	0.11	0.01	0.13	0.01	0.09	0.01	0.83	0.87
CZ-O-P	CZ-O-PT	-0.00	-0.01	0.02	-0.00	-0.00	-0.01	0.32	0.37
CZ-O-PE	CZ-O-PET	0.00	-0.00	0.02	0.01	0.00	-0.00	0.36	0.41
CZ-OT-P	CZ-OT-PT	-0.00	-0.01	0.02	0.00	-0.00	-0.00	0.32	0.35
CZ-OT-PE	CZ-OT-PET	0.00	0.00	0.02	0.01	0.00	0.00	0.36	0.39
CF-O-P	CF-O-PT	-0.00	-0.01	0.02	0.01	-0.00	-0.00	0.33	0.38
CF-O-PE	CF-O-PET	-0.00	-0.01	0.00	-0.00	-0.00	-0.01	0.35	0.40
CF-OT-P	CF-OT-PT	-0.00	-0.01	0.02	0.01	-0.00	-0.01	0.33	0.36
CF-OT-PE	CF-OT-PET	-0.00	-0.01	0.00	-0.00	-0.00	-0.01	0.35	0.38
SZ-O-P	SZ-O-PT	0.00	-0.01	0.01	-0.02	0.01	-0.01	0.83	0.80
SZ-O-PE	SZ-O-PET	0.00	-0.00	0.00	0.00	0.00	-0.00	0.94	0.94
SZ-OT-P	SZ-OT-PT	0.00	-0.02	0.01	-0.01	0.01	-0.01	0.83	0.79
SZ-OT-PE	SZ-OT-PET	0.00	-0.00	0.00	-0.00	0.00	-0.00	0.94	0.92
SF-O-P	SF-O-PT	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.05
SF-O-PE	SF-O-PET	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
SF-OT-P	SF-OT-PT	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.05
SF-OT-PE	SF-OT-PET	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
HZ-O-P	HZ-O-PT	0.02	-0.01	0.04	0.00	0.02	-0.01	0.79	0.78
HZ-O-PE	HZ-O-PET	0.01	-0.00	0.02	0.00	0.01	-0.00	0.87	0.86
HZ-OT-P	HZ-OT-PT	0.02	0.01	0.04	0.00	0.02	0.01	0.79	0.75
HZ-OT-PE	HZ-OT-PET	0.01	0.01	0.02	0.01	0.01	0.01	0.87	0.83
HF-O-P	HF-O-PT	0.03	0.02	0.08	0.01	0.04	0.01	0.41	0.47
HF-O-PE	HF-O-PET	0.06	0.09	0.12	0.07	0.07	0.09	0.33	0.37
HF-OT-P	HF-OT-PT	0.03	0.03	0.08	0.01	0.04	0.03	0.41	0.41
HF-OT-PE	HF-OT-PET	0.06	0.09	0.12	0.07	0.07	0.10	0.33	0.30
B-O-P	B-O-PET	0.11	0.02	0.14	0.02	0.09	0.02	0.81	0.95
B-OT-P	B-OT-PET	0.11	0.01	0.14	0.02	0.09	0.01	0.81	0.86
CZ-O-P	CZ-O-PET	0.02	0.02	0.04	0.03	0.02	0.02	0.30	0.35
CZ-OT-P	CZ-OT-PET	0.02	0.02	0.04	0.03	0.02	0.02	0.30	0.33
CF-O-P	CF-O-PET	-0.01	-0.02	-0.02	-0.01	-0.02	-0.02	0.29	0.35
CF-OT-P	CF-OT-PET	-0.01	-0.02	-0.02	-0.02	-0.02	-0.02	0.29	0.34
SZ-O-P	SZ-O-PET	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.78	0.78
SZ-OT-P	SZ-OT-PET	-0.00	-0.01	-0.01	-0.01	-0.00	-0.00	0.78	0.76
SF-O-P	SF-O-PET	-0.00	-0.00	-0.01	-0.00	-0.00	0.00	0.02	0.01
SF-OT-P	SF-OT-PET	-0.00	-0.00	-0.01	-0.00	-0.00	0.00	0.02	0.01
HZ-O-P	HZ-O-PET	0.00	-0.01	0.01	-0.00	0.00	-0.01	0.69	0.68
HZ-OT-P	HZ-OT-PET	0.00	0.00	0.01	-0.00	0.00	0.00	0.69	0.66
HF-O-P	HF-O-PET	0.03	0.02	0.07	0.02	0.04	0.02	0.41	0.46
HF-OT-P	HF-OT-PET	0.03	0.03	0.07	0.03	0.04	0.03	0.41	0.39

Table A.8: Performance metrics across different models and augmentation strategies