Pós-Graduação em Ciência da Computação

LEVERARING MULTILINGUAL MODELS AND RANK FUSION FOR TRANSLATION MEMORY RETRIEVAL

Por

FILLIPE DE MENEZES CARDOSO DA SILVA

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE/2024

Fillipe de Menezes Cardoso da Silva

LEVERARING MULTILINGUAL MODELS AND RANK FUSION FOR TRANSLATION MEMORY RETRIEVAL

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Luciano de Andrade Barbosa

RECIFE
2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Silva, Fillipe de Menezes Cardoso da.
   Leveraring multilingual models and rank fusion for translation memory retrieval / Fillipe de Menezes Cardoso da Silva. - Recife, 2024.
   95 f.: il.

   Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2025.
   Orientação: Luciano de Andrade Barbosa.
   Inclui referências e apêndices.

   1. Memória de tradução; 2. Neural information retrieval; 3. Rank fusion, ranking; 4. Aprendizado profundo. I. Barbosa, Luciano de Andrade. II. Título.

UFPE-Biblioteca Central

Tese de mestrado apresentada por **Fillipe de Menezes Cardoso da Silva** ao programa de Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **Leveraring Multilingual Models and Rank Fusion for Translation Memory Retrieval**, orientada pelo **Prof. Luciano de Andrade Barbosa** e aprovada pela banca examinadora formada pelos professores:

Prof. Luciano de Andrade Barbosa
Centro de Informática/UFPE

Prof. Vanilson André de Arruda Burégio
Departamento de Computação/UFRPE

Prof. Paulo Rodrigo Cavalin
IBM Research Brazil

RECIFE
2024

I dedicate this work to my family, friends and professors.

Agradecimentos

Inicialmente, gostaria de expressar minha gratidão à minha mãe Taciana e ao meu pai Humberto, por todas as oportunidades que me proporcionaram, por ensinarem-me a relevância da educação, por estarem sempre presentes em todos os momentos da minha vida e pelas lições que me permitiram aprender. Estendo meu agradecimento eterno e sincero aos meus avós, pelo amor e cuidado incessantes. À minha irmã Camilla, pelo compartilhamento de momentos divertidos. À minha tia Thais, que sempre cuidou de mim quando era pequeno.

Ao meu orientador, Luciano Barbosa, pela paciência e dedicação em minha orientação, estou imensamente grato por nossa parceria. Aos demais professores do curso, pelo compartilhamento de seus conhecimentos e experiências. Aos amigos que fiz durante essa trajetória e a aqueles que há muito tempo fazem parte da minha vida.

Não posso deixar de agradecer também à Tayana, pelo seu amor e apoio durante os últimos nove anos, e ao Matheus, por me ensinar que o amor é o combustível da perseverança.

Somewhere, something incredible is waiting to be known.

—CARL SAGAN

Resumo

As memórias de tradução (TMs) são componentes cruciais das ferramentas modernas de Tradução Assistida por Computador (CAT). As TMs armazenam textos traduzidos de uma língua de origem para uma língua de destino, servindo como um repositório de segmentos previamente traduzidos que são essenciais para a produtividade e redução de custos no processo de tradução. No entanto, as bases de dados de TMs atuais dependem principalmente de regras lexicais, como distância de edição ou correspondências de n-gramas, o que limita sua capacidade de identificar traduções semanticamente semelhantes.

Recentemente, os pesquisadores têm explorado o uso de modelos neurais para tarefas de recuperação em TM, porém com um escopo limitado que não aproveita totalmente a natureza multilíngue das TMs e dos modelos e ferramentas neurais disponíveis.

Neste trabalho, exploramos a aplicação de modelos neurais de ponta para o problema de Recuperação de Memória de Tradução e apresentamos nossa pipeline Robust Translation Memory Retrieval (RTMR), que combina modelos neurais e técnicas de recuperação de informações para alcançar resultados de ponta. Além, realizamos amplos experimentos usando uma variedade de TMs e diferentes idiomas de origem e destino, expandindo o escopo de estudos anteriores.

Demonstramos que os modelos neurais não apenas produzem traduções candidatas superiores, mas também oferecem maior flexibilidade e aplicação mais ampla em comparação com abordagens lexicais convencionais. Além disso, mostramos que a integração de técnicas de Rank Fusion (RF) e modelos neurais multilíngues resulta em desempenho de ponta para Recuperação de Memória de Tradução. Nossas descobertas destacam o potencial dos modelos neurais para aumentar significativamente a eficácia dos sistemas de Memória de Tradução.

Palavras-chave: Memória de Tradução, Neural Information Retrieval, Rank Fusion, Ranking, Aprendizado Profundo

Abstract

Translation memories (TMs) are crucial components of modern Computer-Assisted Translation (CAT) tools. TMs store translated texts from a source language to a target language, serving as a repository of previously translated segments that are essential for productivity and cost reduction in the translation process. However, current TM databases rely mostly on lexical rules, such as edit distance or n-gram matches, which limit their ability to identify semantically similar translations.

Recently, researchers have been exploring the use of neural models for retrieval tasks, but with a limited scope that fails to fully leverage the multilingual nature of Translation Memories (TMs) and available neural models and tools.

In this study, we explore the application of state-of-the-art neural models for the Translation Memory Retrieval problem and present our Robust Translation Memory Retrieval (RTMR) pipeline, which combines neural models and information retrieval techniques to achieve state-of-the-art results. Furthermore, we conduct experiments using a wide range of TMs and different languages as source and target, expanding the scope of previous studies that have often been limited to a single TM and language direction.

Through extensive experimentation, we demonstrate that neural models not only yield superior candidate translations but also offer greater flexibility and wider applicability compared to conventional lexical approaches. Furthermore, we show that the integration of rank fusion techniques and multilingual neural models results in state-of-the-art performance for Translation Memory Retrieval. Our findings highlight the potential of neural models to significantly enhance the effectiveness of Translation Memory systems.

Keywords: Translation Memory, Neural Information Retrieval, Rank Fusion, Ranking, Deep Learning

List of Figures

List of Tables

List of Acronyms

Summary

1

Introduction

Translation Memory (TM) is an essential component in today's computer-assisted translation applications. They are essentially databases of pre-translated segments in a particular domain, aimed at streamlining the translation process. According to REINKE (2018), the core concept of TM revolves around its function as a "memory"or cache designed to facilitate future translations. A TM can be viewed as a database that stores segments of text that have been previously translated into one or more languages. These translated segments can then be reused in new translations. The stored text within a Translation Memory can vary in content, size, ranging from small segments to entire paragraphs.

Figure 1.1 presents an illustrative example of a TM. The columns in the figure correspond to different languages, namely English, Portuguese, and Spanish. Each row in the TM contains an identical text unit or segment, translated into the respective languages. These identical text units or segments are referred to as TU. For instance, the texts in the third row, "to the EEA Agreement", "do Acordo EEE", and "del Acuerdo EEE", convey the same meaning despite being in different languages. It is important to notice, a TM does not require translations for all languages covered by a Translation Memory (TM). In fact, it is quite common and more frequently encountered to have translations for only a single language pair within a TU.

Normally, a TM serves as a resource for translation tasks. Given a new source text, the TM system assists the translation process by searching for matching segments within its database. Typically, two primary scenarios arise when attempting to match a new source text with the stored segment texts in a TM:

- Exact Match: The source text aligned precisely, with an entry in the database.
- Fuzzy Match: The TM contains similar segments, but none match the new text completely.

Figures 1.2 and 1.3 presents both scenarios. First, the exatch match sceneario, when the input text has been previously translated and can be directly retrieved from the TM. The second case, the fuzzy match scenario, arises when the input text does not have an exact match within the TM. In this situation, the TM still needs to identify and retrieve the most closely matching segment.

English	Portuguese	Spanish
Decision of the EEA Joint Committee	Decisão do Comité Misto do EEE	Decisión del Comité Mixto del EEE
of 23 April 2004	de 23 de Abril de 2004	de 23 de abril de 2004
amending Annex II (Technical regulations, standards, testing and certification)	que altera o anexo II (Regulamentação técnica, normas, ensaios e certificação)	por la que se modifica el anexo II (Reglamentaciones técnicas, normas, ensayos y certificación)
to the EEA Agreement	do Acordo EEE	del Acuerdo EEE
THE EEA JOINT COMMITTEE,	O COMITÉ MISTO DO EEE,	EL COMITÉ MIXTO DEL EEE,
...	...	...

Figura 1.1: A Translation Memory: An example of a TM encompassing English, Portuguese, and Spanish languages.

The first Translation Memory proposal date back to 1978, due to ARTHERN (1978), "...given portion of text in any of the languages involved can be located immediately, simply from the configuration of the words, without any intermediate coding, together with its translation into any or all of the other languages...". The primary concept was to leverage previously translated content in "controlled"scenarios, as these texts would be considered as grammatically correct and any new similar text to be translated would require only few modifications.

A TM has also proved invaluable even when there is no text to be translated. As highlighted in SOMERS (1997), "an aligned bilingual corpus can also be consulted on a word-by-word basis, where the translator seeks insights into how a particular word or phrase has been previously translated.". Therefore, TMs not only enhance productivity but also elevate the quality of the final translated segments.

TMs are frequently utilized in Computer-aided Translation (CAT) tools, often in combination with other resources such as glossaries and machine translation. A glossary is a comprehensive collection of terms specific to a particular domain, providing definitions for technical terminology and explanations for complex concepts. Additionally, it may include relevant abbreviations, acronyms, and synonyms. Although a glossary can store translations of terms from the source language to a target language, its primary use is to provide definitions and explanations for domain-specific terms. On the other hand, machine translation is a technology that uses algorithms, statistical models or neural models to automatically translate text from one language to another.

Currently, there is an abundance of high-quality Translation Memories (TMs) and parallel corpora available online. These resources are often used in fields like machine translation and other natural language processing tasks. Open source and public TMs can be found from different sources, as government institutions (STEINBERGER et al., 2012), intergovernmental organizations (ZIEMSKI; JUNCZYS-DOWMUNT; POULIQUEN, 2016) and open source pro-

Source	Input Text
English	Decision of the EEA Joint Committee

English	Portuguese	Spanish	Score
Decision of the EEA Joint Committee	Decisão do Comité Misto do EEE	Decisión del Comité Mixto del EEE	100%
of 23 April 2004	de 23 de Abril de 2004	de 23 de abril de 2004	27%
amending Annex II (Technical regulations, standards, testing and certification)	que altera o anexo II (Regulamentação técnica, normas, ensaios e certificação)	por la que se modifica el anexo II (Reglamentaciones técnicas, normas, ensayos y certificación)	14%
to the EEA Agreement	do Acordo EEE	del Acuerdo EEE	59%
THE EEA JOINT COMMITTEE,	O COMITÉ MISTO DO EEE,	EL COMITÉ MIXTO DEL EEE,	68%
...	...	...	...

Figura 1.2: Exact Match: An illustration of an exact match in a Translation Memory, where the input text has a 100% match with a previously translated segment.

jects such as CommonCrawler [1] and ParaCrawl [2]. One notable resource is Opus (TIEDEMANN, 2012), a language resource of parallel corpora and related tools, where a large number of TMs and alignment tools are catalogued, preprocessed and made available for download.

Additionally, a wide range of open-source Computer-Assisted Translation (CAT) tools incorporate TMss into their platforms. Examples include OmegaT (OmegaT Team, 2024), BasicCAT (BasicCat, 2024), and Virtaal (Virtaal, 2024), among others. These tools integrate TMs into the translation process, often complemented by glossaries and Machine Translation (MT) capabilities.

In Figure 1.4, we observe the interface of the OmegaT CAT tool. In the upper left corner, the top-k segments retrieved from the Translation Memory (TM) are presented. In the upper right corner, we see the output of the Machine Translation (MT), and at the bottom is the file being translated.

As mentioned before, a TM can be applied in two distinct scenarios: firstly, for an exact match, where a new source segment aligns perfectly with a previously translated segment; and secondly, a fuzzy match, where a similar segment is retrieved from the TM to aid the linguists in the translation process.

Thus there needs to be a way of assessing the similarity between a new segment and the segments stored in the TM when there is not a 100% match. The usual algorithms applied for this task mostly depend on the overlap of characters as edit distance or some combination of n-gram precision (BLOODGOOD; STRAUSS, 2015). These approaches predominantly rely on lexical matching and tend to favor texts with identical sets of words. Furthermore, it falls short

[1] https://commoncrawl.org/
[2] https://paracrawl.eu/

Source	Input Text
English	Resolution of the EEA Joint Committee 2004

English	Portuguese	Spanish	Score
Decision **of the EEA Joint Committee**	Decisão do Comité Misto do EEE	Decisión del Comité Mixto del EEE	86%
of 23 April **2004**	de 23 de Abril de 2004	de 23 de abril de 2004	49%
amending Annex II (Technical regulations, standards, testing and certification)	que altera o anexo II (Regulamentação técnica, normas, ensaios e certificação)	por la que se modifica el anexo II (Reglamentaciones técnicas, normas, ensayos y certificación)	17%
to **the EEA** Agreement	do Acordo EEE	del Acuerdo EEE	54%
THE EEA JOINT COMMITTEE,	O COMITÉ MISTO DO EEE,	EL COMITÉ MIXTO DEL EEE,	62%
...	...	...	...

Figura 1.3: Fuzzy Match: An illustration of a fuzzy match in a Translation Memory, where the input text does not have a 100% match with any previously translated segment. The TM should retrieve the closest matching segment(s)

in recognizing semantic similarities between segments that may encompass different wordings or tokens, yet convey a similar meaning.

Recently, Deep Learning (DL) has led to significant advancements in fields such as machine translation, computer vision, and speech recognition. This success has also prompted the employment of neural models to generate improved semantic representations of text inputs, models such as Universal Sentence Encoder (USE) (CER et al., 2018), LaBSE (FENG et al., 2020), and more (REIMERS; GUREVYCH, 2019). These models produce dense representations of texts in embedding space, where semantically similar sentences are positioned close together, and dissimilar sentences are positioned farther apart. Many of these models are multilingual, meaning that similar texts in different languages are also positioned close together, while dissimilar texts are positioned farther apart. The dense vector generated by these models enables more accurate comparisons of textual data based on their semantic meaning in a multilingual scenario.

Regarding the TM Retrieval problem, RANASINGHE; ORASAN; MITKOV (2020) have previously applied neural models, specifically the Universal Sentence Encoder (USE). However, their approach has a limited scope as it only utilizes source segments for comparison, which does not fully leverage the multilingual data within TMs and available multilingual neural models and tools. Furthermore, many studies have evaluated TM Retrieval only for a specific language pair, a particular TM, and using lexical evaluation rules that may not be suitable for all cases (KIM et al., 2021), (CAI et al., 2021). In contrast, our work aims to address these limitations by exploring a wider range of TMs, language pairs, and semantical evaluation metrics.

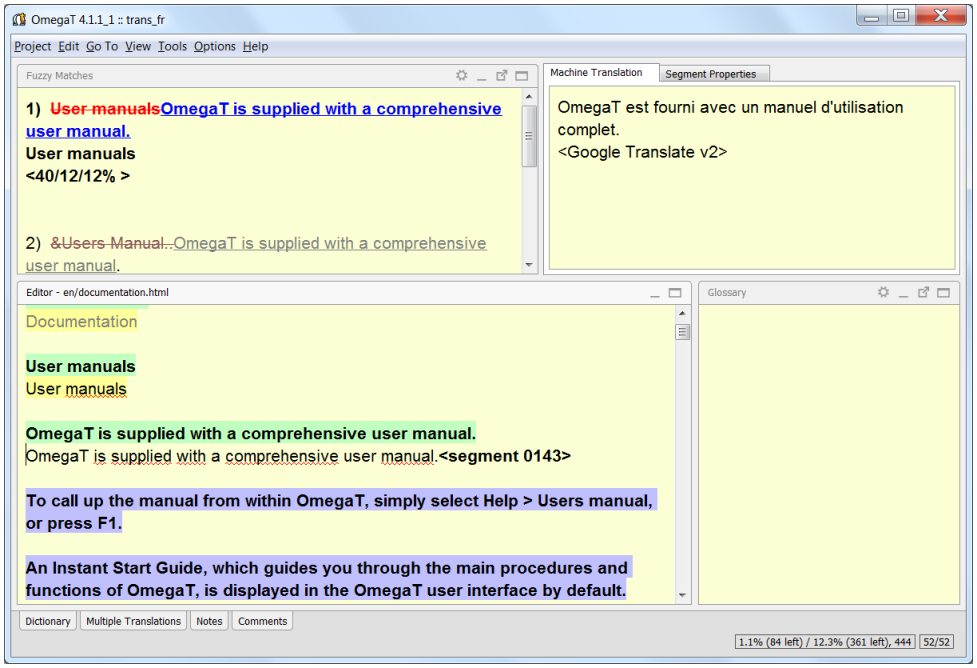
OmegaT 4.1.1_1 :: trans_fr
Project Edit Go To View Tools Options Help
Fuzzy Matches
1) User manualsOmegaT is supplied with a comprehensive user manual.
User manuals
<40/12/12% >
2) &Users Manual..OmegaT is supplied with a comprehensive user manual.
Machine Translation
Segment Properties
OmegaT est fourni avec un manuel d'utilisation complet.
<Google Translate v2>
Editor - en/documentation.html
Documentation
User manuals
User manuals
OmegaT is supplied with a comprehensive user manual.
OmegaT is supplied with a comprehensive user manual.<segment 0143>
To call up the manual from within OmegaT, simply select Help > Users manual, or press F1.
An Instant Start Guide, which guides you through the main procedures and functions of OmegaT, is displayed in the OmegaT user interface by default.
Glossary
Dictionary
Multiple Translations
Notes
Comments
1.1% (84 left) / 12.3% (361 left), 444
52/52

Figura 1.4: OmegaT: An open source CAT tool with Machine Translation and Translation Memory.

1.1 Problem Definition

A general definition of the Transltion Memory Retrieval problem can be given as follows: Starting by defining the small individual components of a TM the Translation Unit (TU) (GALA Global., 2024). Each TU consists of segments in various languages that convey the same meaning or can be deemed equivalent. A Translation Memory is then simply defined as a collection of TUs. In addition we can refer to TM as bilingual, if there is only a source and a target language, Or multilingual if it encompasses more than two languages.

Translation Memory Retrieval is the process of finding matching or similar translations to a given segment in a Translation Memory (TM). When presented with a new source segment, with its language and a desired target language, a well-designed TM system would retrieve the most closely corresponding translation accessible for the specified target language.

In a formal setting, let us represent a translation unit as $TU_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$, with $L = 1, \ldots, N$ languages, where each $x_{i,j}$ corresponds to a segment in a specific language j.

So given a TM of size M, a new source segment to be translated x, a desired target language, the objective is to retrieve the most similar translation unit TU_j from the TM. Usually this is achieved by employing a similarity function, denoted as $f(x, TU_i)$, which quantifies the degree of similarity between the source segment x and some segment from the translation unit TU_i. We can summarize this in the following equation 1.1.

$$TU_j = \underset{1 \ldots M}{\operatorname{argmax}} f(x, TU_i) \tag{1.1}$$

Here, it's important to highlight that the selection of a similarity function has a big

influence on the results. The nature of f, whether it employs a lexical or semantic approach and whether languages beyond the source language are involved, can lead to substantial result variations.

Translation Memory Retrieval falls into the symmetric search category. In symmetric search, there is a notable similarity between the indexed texts and the query texts in terms of content and length. Translation memories serve as a good example of this category, where the indexed data comprises translated texts, and queries typically involve pieces of text to be translated. Conversely, asymmetric search presents a different dynamic. For instance, queries in a search engine exemplify asymmetric search, where the query itself tends to be concise, reflecting the user's needs, while the resulting answers are typically lengthy texts.

1.2 Main Objectives

In this work, we delve in the application of Neural Information Retrieval (Neural IR) for the task of Translation Memory Retrieval. For this, we explore a spectrum of approaches, ranging from baselines implementations of dense retrievers to use of multilingual models with rank fusion techniques. We've also expanded the common evaluation framework to incorporate multiple TMs, with different source and target languages directions, and a more appropriate evaluation metrics that takes into account the semantic similarities.

Here is a short list of our main contributions for the task of Translation Memory Retrieval:

- Baseline Models Comparison: We evaluate a wide array of baseline approaches for the first stage ranking, including BM25, Universal Sentence Encoder, LaBSE, ME5-Multiligual, and others open source models, and more.
- Second stage ranker: We believe to be the first to assess the effectiveness and challenges associated with reranking for the task of Translation Memory Retrieval.
- Appropriate Metrics for Evaluation: We choose neural metrics for assessing the performance of TM Retrieval. Previous works only focus on lexical metrics as the METEOR, which we find not suitable for the problem.
- Incorporating Multilingual Texts: We delve into harnessing the multilingual aspect of Translation Memories, specifically leveraging multilingual texts to enhance retrieval results.
- Rank Fusion for TM Retrieval: We employed rank fusion to enhance the performance of Translation Memory Task retrieval.

2

Foundations

This chapter introduces the fundamental concepts underlying our work. Firstly, we provide an overview of ranking strategies, starting with traditional information retrieval approaches and progressing to more recent neural models. Specifically, we delve into the Transformer neural architecture and its today's significance for information retrieval and neural search. Finally, we provide detailed explanations of the translation metrics utilized in this work, highlighting their relevance and applicability to our research objectives. By establishing a strong foundation in these key areas, we aim to provide a comprehensive understanding of the methods and techniques that underpin our work.

2.1 Information Retrieval

Information Retrieval (IR) as a research field is defined by SCHÜTZE; MANNING; RAGHAVAN (2008) as: "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)". Typically, users search for stored information in a database, and in contrast with relational systems, where data is stored in a well-structured format, in IR systems, data is stored in an unstructured manner, often in various textual formats such as paragraphs, documents, web pages, and more. The IR system is tasked with locating and delivering the most pertinent content from the database to the user.

For most IR systems, the users' needs or desires are also expressed in a plain text format, the query. GARCIA (2009) points out that the query serves as a representation of an individual user's information need or search intent. However, this definition alone does not guarantee its adequacy or comprehensiveness.

So given a query and a database of stored documents, we need a way to evaluate the relevance of user queries to the stored documents, it is necessary to establish a method for comparison. The following section will discuss traditional techniques for transforming both the query and the documents to enable relevance assessment. Subsequently, we will explore more recent approaches to further enhance this process.

2.1.1 Vector Space Model

The Vector Space Model (VSM) encapsulates queries or documents within a multi-dimensional vector or embedding. Each dimension of the vector corresponds to a specific word or term from the vocabulary, with the value of each dimension indicating the term's significance within the text. The relevance of a document for a query is calculated as the distance between the angles, cosine similarity 2.1, of the query embedding and document embedding. At Figure 2.1 we can see an example of a query q and two documents d_1 and d_2. Document embeddings that assign greater significance to terms present in the query are more likely to be ranked higher compared to those that do not prioritize these terms to the same extent.

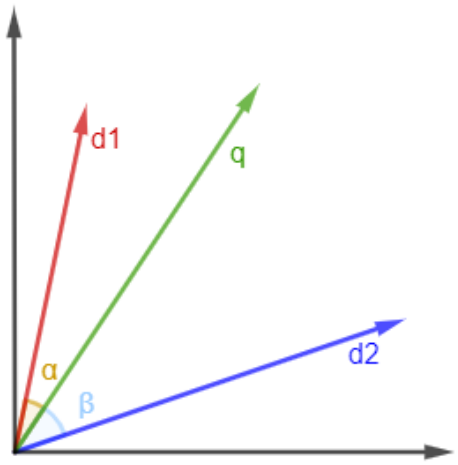

Figura 2.1: Query and document example in a vector space model.

$$cos(a,b) = \frac{< a.b >}{||a.b||} \tag{2.1}$$

2.1.1.1 Term Frenquency

A more plausible approach to give involves determining the weights of each dimension based on Term Frequency (TF). Term frequency $\mathrm{tf}_{\mathrm{t,\,D}}$ is computed by counting the occurrences of a term t within a document D and dividing it by the total number of times the terms appears in the text. TF gives more importance to terms that occur frequently within a document.

The weight for a term t for a document D with N terms is defined in the equation 2.2.

$$\mathrm{tf}_{\mathrm{t,\,D}} = \frac{\text{Number of times term } t \text{ appears in document } d}{\mathrm{N}d} \tag{2.2}$$

Some $\mathrm{tf}_{\mathrm{t,\,D}}$ variations are shown in Table 2.1:

2.1.1.2 Inverse Document Frequency

Inverse Document Frequency (IDF) is another way of assessing the terms relevance in IR. It measures how rare or common the term is across the entire collection of documents. So

Name	Name	Term Frequency
TF	Log Normalization	$1 + log(\text{tf}_{\text{t, D}})$
TF	Double Normalization K	$0.5 + 0.5 * \frac{\text{tf}_{\text{t, D}}}{\max_i \text{tf}_{\text{t, D}}}$
IDF	Inverse Frequency Smooth	$\log\left(1 + \frac{\text{N}}{n_t}\right)$
IDF	Inverse Frenquecy Max	$\log\left(1 + \frac{\max_i n_i}{n_t}\right)$
IDF	Probabilistic Inverse Frequency	$\log\left(1 + \frac{N - n_i}{n_t}\right)$

Tabela 2.1: TF and IDF variants.

given the number of documents in the collection N, and n_t the number of documents that the term t appears, the IDF is calculated as follows 2.3:

$$\text{idf}_{\text{t, D}} = \log\left(\frac{\text{N}}{n_t}\right) \tag{2.3}$$

Some $\text{idf}_{\text{t, D}}$ variations shown in Table 2.1:

2.1.1.3 TF-IDF

Integrating both Term Frequency (TF) and Inverse Document Frequency (IDF), we calculate the Term Frequency-Inverse Document Frequency (TF-IDF) score for each term t as represented by 2.4. This captures the significance of a term by considering both its frequency and rarity within a corpus. Its one of the most common and used methods for assessing the relevance of terms in IR BAEZA-YATES (1999).

$$\text{TF-IDF(t, D)} = TF(t,D) \text{ x } IDF(t,D) \tag{2.4}$$

2.1.1.4 BM25

BM25 is a probabilistic retrieval model that is widely used in information retrieval and text mining applications (ROBERTSON; ZARAGOZA et al., 2009). It is based on the Probabilistic Relevance Framework (PRF), which models the relevance of a document given a query as a probability $P(d|q)$.

Given a document D and a query Q the BM25 relevance score is calculated as follows:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1 + 1)}{f(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \tag{2.5}$$

Where:

- q_i: The ith query term.
- $f(q_i,D)$: Term frenquency in the document.

- $IDF(q_i)$: IDF for the q_i term.
- $\frac{|D|}{\text{avgdl}}$: The length of the document divived by the average length in the database.
- b: Influences how much the ratio of the length of the documents affects the final score. If set to zero, the effects of the ratio length would be canceled.
- k_1: A parameter that limits how much a single query term can affect the final score.

2.1.2 Learning to Rank

In contrast to previous methods for ranking and assessing the relevance of query and documents, LTR techniques leverage the supervised Machine Learning (ML) framework ZHOU (2021) to help improve the relevance of search results or recommendations. LTR techniques use data to train a function that directly ranks documents based on their relevance to a given query.

At Figure 2.2 we have a a full view of the LTR framework, given a dataset of queries and features of documents, the LTR employs classical machine learning techniques to train a model that accurately predicts the relevance ranking of queries and documents. The model is first trained on a train dataset and subsequently evaluated on a separate test dataset to assess its performance on unseen data.

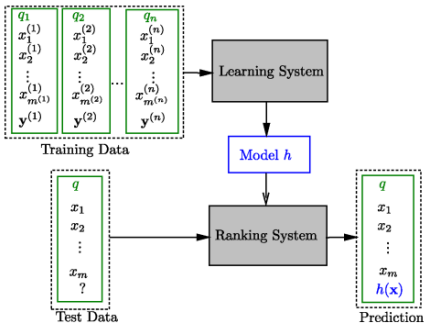

Figura 2.2: Learning to Rank Framework. Source LIU et al. (2009)

According to LIU et al. (2009), LTR approaches can be classified in:

- Pointwise: In this approach, the input to a LTR model consists of a feature vector derived from both the query and the document. The model is trained to compute the relevance of a document to a given query based on its feature vector.
- Pairwise: In contrast, the pairwise approach takes pairs of feature vectors representing two different documents. The model's output indicates the preference between these two documents for a particular query. By comparing pairs, the model learns to rank documents in the order of their relevance to the query.

- Listwise: In this methodology, the input comprises a list of documents along with their corresponding feature vectors, representing various possible rankings of documents for a given query

LTR models are commonly used alongside previous approaches like VSM or BM25 in a multi-stage ranking pipeline (ANAND et al., 2021), (HAN et al., 2020). A VSM as TF-IDF with cosine or BM25 would serve as a first-stage ranker, fast filtering and returning the top-k segments from the index. Subsequently, an LTR model is employed to further refine the output.

2.2 Neural Information Retrieval

Neural IR is the application of neural networks to common IR tasks (MITRA; CRASWELL et al., 2018), like assessing a query and document relevance, semantic first-stage retrieval, question answering, recommender system and others. In the following section, we will focus on two primary Neural IR approaches: the Cross Encoder and the Bi-Encoder.

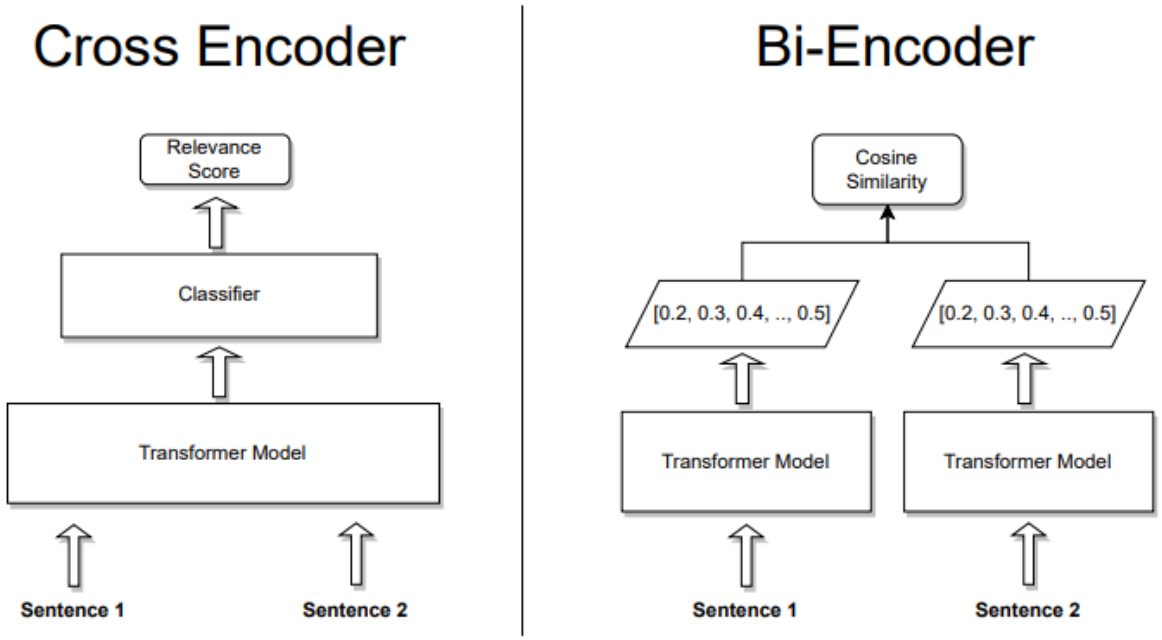

Figura 2.3: Cross encoder schema vs Bi-encoder schema

2.2.1 Cross Encoders

A cross-encoder model takes two sentences as input and computes the relevance score between them. Given a query sentence q and a document sentence d, a cross-encoder model f outputs the similarity of q and d as $f(q,d) \in (0,1)$, where values closer to zero indicate dissimilarity between q and d, while values closer to 1 signify a high degree of similarity.

Cross encoders are a type of pointwise LTR, that takes the text input directly instead of a feature vector. Therefore, it requires a training dataset before it can be effectively employed for

downstream tasks. Although there are pre-trained cross-encoders available that can be used out of the box (REIMERS; GUREVYCH, 2019), it is important to note that they may not always deliver optimal performance for a specific domain.

2.2.2 Bi-encoders

Bi-encoders, also known as dual encoders or sentence embedding models, are a class of neural network architectures that are commonly used in natural language processing tasks such as semantic search and text classification. Given two sentences x and y, a bi-encoder model works by producing a dense representation of the input text in embedding space, where semantically similar sentences are positioned close together and dissimilar sentences are positioned farther apart. This type of models have been widely use for bi-text mining (GUO et al., 2018), translation memory retrieval (RANASINGHE; ORASAN; MITKOV, 2020), passage retrieval (LU et al., 2022) and more.

Figure 2.3 shows the framework of using a bi-encoder model for semantic search. In this approach, two sentences are passed through the same transformer model, which outputs a dense embedding representation of the text. The embeddings are then compared using a similarity function, such as cosine similarity or dot product, to determine the semantic similarity between the two sentences.

2.2.3 Rank Fusion

Rank Fusion (RF) is a technique that combines multiple ranked lists, typically from different ranking methods, into a single merged rank. Rank fusion aims to create a new rank that outperforms individual ranks by combining their unique strengths and representations in matching queries to documents, and thus resulting in a more accurate ranking.

The CombSUM method is a simple rank fusion technique used for combining results from multiple searches (FOX; SHAW, 1994). It relies on the score output of each document in each rank. The new rank is formed by summing the individual scores for each document in each rank into a final score. Given a list of ranks R the CombSUM scores a document d as follows:

$$\text{CombSUM}(d) = \sum_{r \in R} Score(d, r) \tag{2.6}$$

Where Represents the set of all ranks being combined, r represents an individual rank in R, and Score(d, r) represents the score of the document d in rank r. The CombSUM method sums up the scores of document d across all ranks in L to produce a final score, which is used to rank the documents in the merged list.

Reciprocal Rank Fusion (RRF) is a rank fusion technique that shows better performance when compared to other techniques (CORMACK; CLARKE; BUETTCHER, 2009). Instead of using the individual scores, the RRF leverages the rank position of a document in each rank to

generate a new rank score.

Given a set of documents D and a list of rankings R, the RRF score is computed as follows:

$$RRF(d \in D) = \sum_{i}^{n} \frac{1}{k + r(d)} \tag{2.7}$$

Where n is the number of rankings for merging, k is a free parameter that weights the importance of the document position. A higher value of k will result in a lower importance being assigned to the document's rank, while a lower value of k will assign greater importance to the document's rank. The RRF score is determined by calculating the multiplicative inverse of the rank position. The method accumulates the reciprocal rank scores for each document across all the input rankings, giving greater weightage to documents that appear at the top of multiple rankings, while still taking into account the rank positions of all documents in the input rankings.

2.3 Foundation Models

In recent years, significant advancements have been made in fields such as machine translation, computer vision, speech recognition, and many others (LECUN; BENGIO; HINTON, 2015). Many of these breakthroughs stem from progress in training new architectures of neural networks, particularly Deep Neural Networks (DNN), which are characterized by their deep architectures, consisting of multiple layers that when trained with enough data can discovers intricate patterns. Nowdays networks are generally trained with the self-supervised learning paradigm JAISWAL et al. (2020), which involves generating pseudo-labels for training.

Recently, the Transformers architecture (VASWANI et al., 2017) has been sucefully use and deploy for problems that involves text. The original architecture shown in Figure 2.4, consists of two main parts, the encoder and decoder. In the encoder all tokens attend to one another, allowing each token to consider information from all other tokens in the input sequence. Conversely, in the decoder block, tokens can only attend to preceding input tokens.

This transformers models have also been called as Foundation Models: "A foundation model is any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks..."BOMMASANI et al. (2021). These models show remarkable adaptability and performance when fine-tuned for specific tasks, including scenarios where there is limited labeled data for the final tasks. This adaptability comes from their understanding of underlying patterns and structures of texts they gained during the pre-training. The whole evolution of AI paradigma is shown at figure 2.5.

BERT (Bidirectional Encoder Representations from Transformers) is one of the most known and used foundation models. It achieved many state of the art results for natural language tasks like, question answering, text classification and more (DEVLIN et al., 2018). At the core of BERT's functionality is its ability to generate contextually rich representations of words, This is

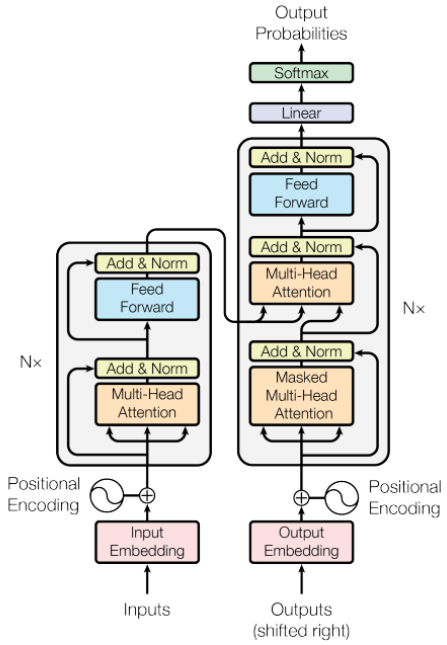

Figura 2.4: The Transformer architecture, by (VASWANI et al., 2017).

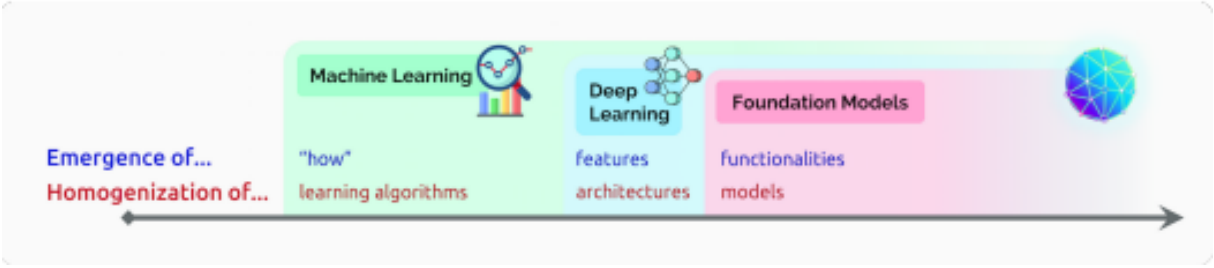

Figura 2.5: The evolution of the AI paradigma. Source BOMMASANI et al. (2021)

achieved through the use of transformer encoder layers, where each layer of the encoder process input text in a way that allows each token to attend to all other token in the sentence, therefore capturing the complex linguistic patterns and dependencies.

When applying BERT to specific downstream tasks, the input text is first tokenized and then formatted in a task-specific manner. For instance, in the case of comparing the relevance of two texts x and y, the input is formatted as $[CLS] + x_1, ..., x_n + [SEP] + y_1, ..., y_p$, where $[CLS]$ is a special token representing the beginner of a sentence, $[SEP]$ a special token for BERT used to separate two sentences and $x_1, ..., x_n$ the list of tokens from sentence x and $y_1, ..., y_p$ the list of tokens from sentence y.

For downstream tasks such as text classification or sentence pair classification, the embedding corresponding to the $[CLS]$ token is typically used as input to a task-specific head. The input is formatted as follows:

$$\text{input} = [x_1, \ldots, x_n] \text{ and } [y_1, \ldots y_p]$$
$$\text{embeddings} = \text{BERT}([CLS], x_1, \ldots, x_n, [SEP], y_1, \ldots, y_p)$$

The embedding corresponding to the $[CLS]$ token is then passed through a feedforward neural network with a softmax activation function to produce a probability distribution over the possible output labels. This can be represented as:

$$\text{output} = \text{softmax}(W \cdot [\text{CLS}]_{embedding} + b) \tag{2.8}$$

Figure 2.6 shows the input and output of a BERT model for the classification problem.

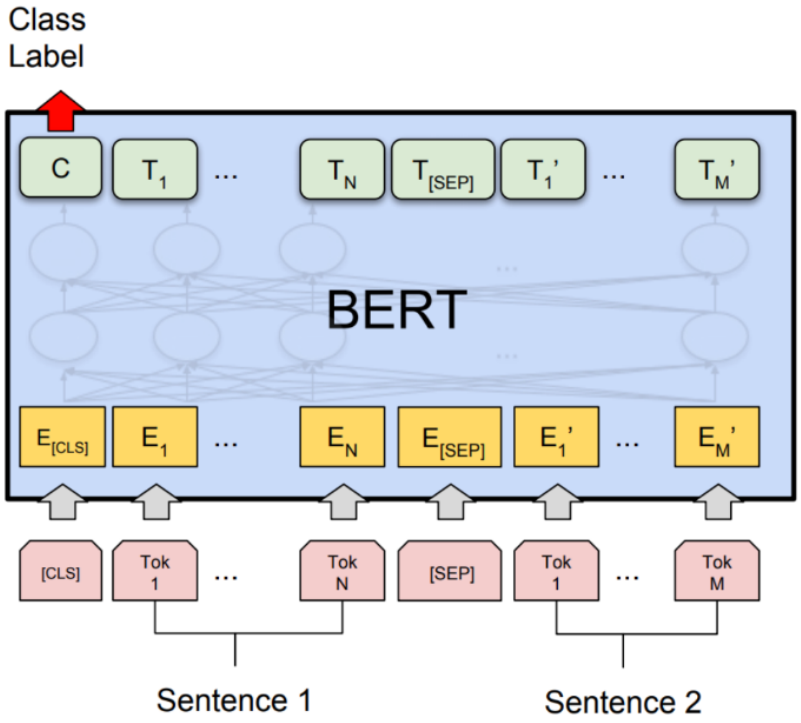

Figura 2.6: BERT Architecture: A simplified view of the BERT architecture. At the bottom, the two sentences are first tokenized. Then, the initial embeddings are retrieved for each token, which then pass through layers of the BERT model. The output consists of embeddings of the same size as the input. Typically, a language head model is attached to the $[CLS]$ token to provide a way of classifying examples.

2.4 Machine Translation

Machine Translation refers to the use of statistical or machine learning approaches to automatically translate texts or speech from one language to another. These approaches

utilize algorithms and models to analyze and understand the source language and generate a corresponding translation in the target language.

Over the years, MT has been studied within different paradigms (GARG; AGARWAL, 2018). Initially, rule-based approaches were used, which relied on morphological, syntactic, semantic, and knowledge about the source and target sentences to perform the translation (SHIWEN; XIAOJING, 2014).

Following we had Statistical Machine Translation (SMT) approaches, where it is assumed that every source sentence S has a possible translation T. SMT assigns a probability $P(T|S)$ for every sentence pair from the source and target language. So the statistical approach defines the goal of finding the sentence T that maximizes this probability. SMT is divided into Word-based and Phrase based, more details in LOPEZ (2008).

Currently, DL based neural networks have been the state-of-the-art for performing translation tasks. The Transformer architecture itself was first applied for the translation task and achieved state-of-the-art results for the translation of texts from English-to-German (VASWANI et al., 2017). In this scenario, encoder-decoder transformers are typically trained on parallel datasets to learn how to map the initial sentence to the translated one.

MT research has focused on a set of points: the training and releasing open-source state-of-the-art models for specific language pairs (ÖSTLING et al., 2017). On the ability to translate between any language pairs without the need to use English as an intermediate language (TANG et al., 2020) and extending models to handle languages with limited bi-text data available (COSTA-JUSSÀ et al., 2022). Moreover, there has been a growing interest in developing multimodal models that can translate speech-to-speech, text-to-speech, and text-to-text, such as the SeamLessM4T model from BARRAULT et al. (2023).

2.5 Machine Translation Metrics

Machine translation systems can be evaluated through human evaluation or automated evaluation. However, human evaluation is a costly and time-consuming process PAPINENI et al. (2002) that primarily relies on a group of linguists to assess the quality of a machine translation system. Moreover, human evaluation can be highly subjective since it depends on individual judgments of what constitutes a good or bad translation (LEE et al., 2023). Additionally, it can be challenging to apply human evaluation to different domains or systems as it requires manual work. On the other hand, automated evaluation metrics have a very low cost and can be used to quickly compare multiple machine translation systems.

MT metrics were initially designed to compare translated segments with reference translations using some lexical matching rules to generate a translation quality score. Similar to traditional information retrieval (IR) methods and Neural IR, these lexical MT metrics rely on matching subparts of the translated target with a reference. However, they tend to fall short in capturing more nuanced, semantic meanings. In other words, these metrics may not fully

account for the meaning of the text beyond simple word-for-word matching.

With the advent of new methods and models from deep learning, discussed in section 2.3, there are now metrics available that are more semantically oriented and can provide evaluations that are closer to human judgments.

Below, we list a set of very common traditional metrics for evaluating machine translation (MT) systems, including lexical metrics like BLEU and chrF, as well as metrics that attempt to capture more semantic meaning from the text like METEOR. In addition, we will also discuss newer learned approaches that use neural networks, such as BLEURT and COMET.

2.5.1 BLEU

The BLEU (Bilingual Evaluation Understudy) (PAPINENI et al., 2002) is one of the most widely used metrics for evaluating the quality of machine translation output. It is based on a modified n-gram precision, which counts the number of times an n-gram appears in the candidate translation and in the reference translation. To avoid a MT system producing an excessively long translation, the total count of n-grams in the candidate translation is clipped by the number of n-grams in the reference translation.

Specifically, BLEU is defined as as bellow:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.9}$$

Where N is the maximum n-gram order, normally up to 4, w_n is the weight for each n-gram, typically set to $1/N$, and BP (Brevity Penalty) is a factor for discouraging short sentences.

BLEU is a purely lexical metric, meaning that it only considers exact matches of n-grams between the candidate translation and the reference translation. It does not take into account any kind of semantic or syntactic similarity between words. Therefore even if a translation uses synonyms or the sentence has the same meaning as the reference, BLEU could not give credit for these matches. BLEU was also designed to evaluate the quality of MT systems on a corpus level, although there are some variants that use smoothing techniques to address the issue of low n-gram counts (CHEN; CHERRY, 2014).

2.5.2 chrF

The chrF metric computes the F-score based on the overlapping n-grams found in the source text and the reference text. According to (POPOVIĆ, 2015), the chrF Score is defined as:

$$chrF = (1+B^2)\frac{chrP * chrR}{B^2 * chrP * chrR} \tag{2.10}$$

where *chrP* represents the percentage of n-grams in the source text that have a matching counterpart in the reference text, and *chrR* denotes the percentage of n-grams found in the reference text that also appear in the source text.

B defines how much importance we give for *chrP* and *chrR*, a higher *B* gives more importance to *chrR* and a lower *B* more importance to *chrP*.

chrF does not rely on the input texts to be tokenized, as it defines each n-gram as a sequence of characters, normally a sequence of 6 characters, rather than as a sequence of words. This makes chrF a more stable metric for languages where there are no clear or defined word boundaries, such as Chinese or Japanese.

2.5.3 METEOR

METEOR (BANERJEE; LAVIE, 2005) works by creating an alignment between the candidate text and reference text, where each alignment is a mapping between unigrams from the candidate translation to the reference translation. In this mapping, a single candidate unigram can only map to another reference unigram or remain unmapped. METEOR evaluates a candidate translation by computing a score based on the harmonic mean of precision and recall of the correct unigram matches between the candidate and reference texts, with a penalty factor applied given the final alignment. The mapped unigrams are grouped into the fewest number of chunks, where a chunk is as a continuous sequence of unigrams that are mapped to the same continuous sequence of unigrams in the reference text.

The score is calculated as follows:

$$\text{Penalty} = 0.5 * (\frac{\text{chunks}}{\text{unigrams matched}})^3 \tag{2.11}$$

$$\text{METEOR} = \frac{10PR}{R+9P} * (1 - \text{Penalty}) \tag{2.12}$$

The alignment process in METEOR consists of three main steps. First, the candidate and reference texts are tokenized into individual words, and exact matches between the two texts are aligned. In the second step, the remaining words are stemmed using a Porter Stemmer-type algorithm, which reduces each word to its base form. The stemmed words are then mapped between the candidate and reference texts. Finally, in the third step, any remaining unmapped words are matched using WordNet, a large lexical database for English that groups words into sets of cognitive synonyms, the synsets. If the remaining words belong to the same synsets, they are also mapped.

METEOR . was developed to address some of the weaknesses of previous metrics, such as the BLEU, which rely on the exact matches between n-gram from the translation text and the reference text (BANERJEE; LAVIE, 2005). However, as we can see METEOR still depends on the availability of specific tools, such as tokenizers and WordNet, to align words with the same base form and to match related words. These tools may not be available or may not work well for all languages, which can limit the effectiveness of METEOR for evaluating translations in certain language pairs.

2.5.4 BLEURT

BLEURT is a learned evaluation metric based on the BERT transformers architecture SELLAM; DAS; PARIKH (2020). Given a candidate translation $x = (x_1, ..., x_r)$ of length r with each x_i is a token, a reference translation $\hat{x} = (\hat{x_1}, ..., \hat{x_p})$ of length p, BLEURT team trained a BERT model to to predict the human quality score of the translatation $f : (x, \hat{x}) -> y$. When $y \in [0, 1]$, a value close to zero signifies poor translation quality, whereas a value closer to one indicates high translation quality.

BLEURT model was trained initially on a large synthetic data of translations, built but perturbing 1.8 million segments from the Wikipedia with back translation, mask-filling and randomly dropping out words. After pre-training the model was fine tuned with the WMT Metrics Shared Task data.

BLEURT also comes with a version trained with the RemBert, a transformer model (CHUNG et al., 2020), fine-tuned on the WMT Metrics Shared Task data collected between 2015 to 2019 and a synthetic data with 160K sentence pairs derived from the WMT corpus.

There are number of other BLEURT versions and checkpoints available online. Table 2.2 presents details about them (PU et al., 2021). The BLUERT-20 checkpoint emerges as the superior model and is the recommended model by the authors to run and report experiments.

Model	Agreement w. Human, to-En.	Agreement w. Humans from-En.	Parameters
BLEURT-20	0.228	0.520	579M
BLEURT-20-D12	0.219	0.467	167M
BLEURT-20-D6	0.211	0.429	45M
BLEURT-20-D3	0.191	0.385	30M

Tabela 2.2: BLEURT Model Performance Metrics

1

3

Background

This chapter showcases two primary illustrations of how our solution integrates into the translation pipeline. By examining these cases, we aim to provide a clear context for the operation Robust Translation Memory Retrieval (RTMR) solution, and to highlight its significance.

3.1 Traditional Scenario

In a typical TM application, translators use CAT tools with integrated TM systems to translate documents (GARCIA, 2009). The CAT tool searches the TM for matching segments, offering exact or fuzzy matches to the translator. The translator then accepts, edits, or rejects the suggested translations. The updated translations are saved back in the TM, enabling more efficient and consistent translations.

In this scenario, our proposed RTMR solution will replace the matching process from the default TM system, offering improved translation suggestions and enhancing the overall efficiency and quality of the translation process within the CAT tool.

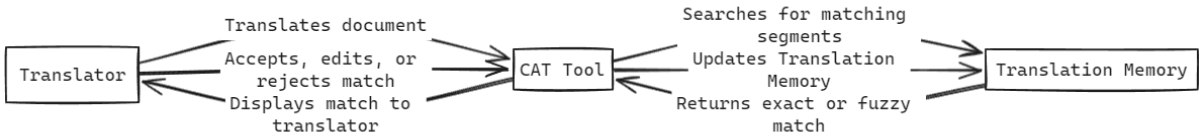

Figura 3.1: Translation Memory Traditional Use Case: The figure shows the collaboration between linguist, CAT Tool, and TM in the Translation Process.

3.2 Translation Memories for In-context learning

Translation Memories can also be utilized for in-context learning in Large Language Model (LLM) generation tasks, particularly for translation. By providing similar text segments, TMs can assist LLMs in generating improved translations (LIU et al., 2021). Our proposed RTMR solution will replace the matching process to find the top-k siilar segments, instead of returning only the most similar example. Figure 3.2 details how a TM, can be integrate into this pipeline:

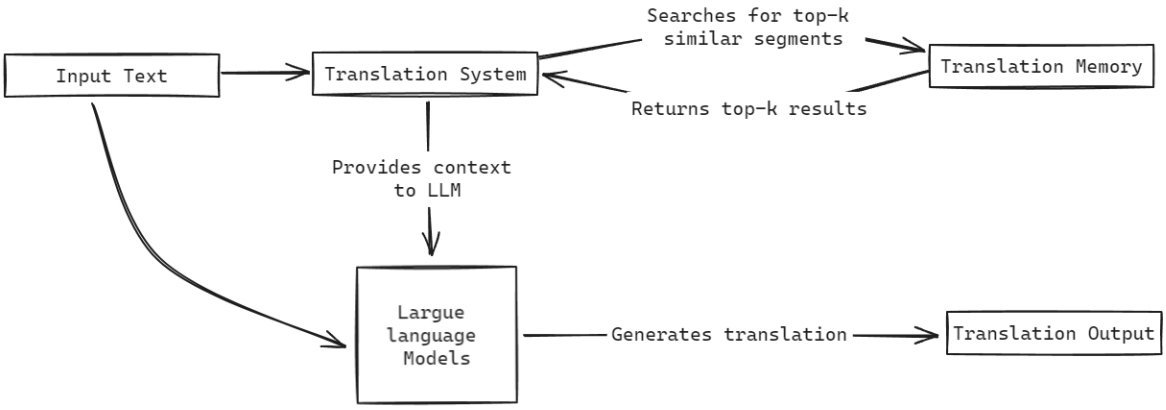

Figura 3.2: Translation Memory used for In-context Learning: The illustration demonstrates how TM results is integrated into the generation process of LLM, for the translation task.

4

Related Work

In this chapter, we provide a comprehensive review of previous works related to translation memory retrieval, as well as those in other fields where a translation memory retrieval pipeline is a component of the larger system. We highlighted the similarities and differences between these prior works and our own research. We divided the next sections into two main categories: Lexical and Semantic approaches. Lexical Approaches will typically rely on more traditional and lexical rule-based ranking systems and Semantic Approaches tend to leverage foundational models for retrieval.

4.1 Lexical Approaches

KIM et al. (2021) address the challenge of Translation Memory Retrieval by focusing on efficient retrieval methods. They employ the vector space model (VSM) along with a WordNet type query expansion technique to try to handle the issue of matching sentences with similar meanings but different vocabulary. Given a TM they propose the following matching algorithm. First they define a V matrix as the word-sentence relation, where the ith row is representing a source segment $V_{S_i} = (v_{i1}, v_{i2}, ..., v_{iT})$, and each v_{ij} indicates the importance of a word to the source segment. The input sentence is defined as $U_{s0} = (u_1, u_2, ..., u_r)$, with each u_j indicating the weight of each word for the input sentence. Then given two vectors from V_{S_i} and U_{s0} their similarity is simply the cosine from the two vectors.

The weights v_{ij} are computed as the minimum term frequency of the word w_j between the indexed source sentences S_i and the input sentence U_{s0}. This is formally represented as:

$$v_{ij} = \min\{tf(w_j, S_0), tf(w_j, S_i)\}$$

For u_i is assigned the inverse document frequency score of the term w_j if the term is present in the source sentence U_{s0}. Otherwise, it is assigned a value of 0. This can be formally written as:

$$u_i = \begin{cases} \text{idf}(w_j), & w_j \in S_0 \\ 0, & w_j \notin S_0 \end{cases}$$

To account for semantically similar translations that might use different vocabulary but convey similar meanings, the authors expanded the input sentence query with the help of WordNet FELLBAUM (1998). For each word w_j in the source sentence U_{s0}, all other words in the same synset as w_j are assigned a value of α, where α is a real number between 0 and 1. This can be formally written as:

$$v_{ij} = \begin{cases} \alpha, & \text{if } w_j \in \text{synset}(w_k) \text{ for some } w_k \in U_{S_0} \\ \min\{tf(w_j, S_0), tf(w_j, S_i)\}, & \text{otherwise} \end{cases} \tag{4.1}$$

The strategy of expanding words using Wordnet, has certain limitations. This approach does not take into account the contextual meaning of words and remains dependent on language-specific tokenizers.

The researchers in (GUPTA et al., 2016) utilized a hybrid approach that integrated edit distance and paraphrasing techniques to detect and retrieve comparable segments. To prevent the application of edit distance to an extensive set of paraphrases that could potentially become unmanageable, they leverage dynamic programming and greedy algorithms. They used the PPDB 1.0 paraphrases database (GANITKEVITCH; VAN DURME; CALLISON-BURCH, 2013), which contains over 220 million paraphrases pairs for the English language. In order to mitigate the cost of applying edit distance to a large group of paraphrases, they categorized the paraphrases into four distinct groups, followed by the application of filters to remove sentences that could negatively impact the edit distance algorithm's speed. In contrast to the methodology presented in (KIM et al., 2021), where word representation was altered according to the synset group, this approach utilizes paraphrasing while maintaining edit distance as the lexical relevance function. The approach also depends on available paraphrases in the database, which may limit its applicability in certain scenarios. For instance, when dealing with Translation Memories in specialized domains or containing lengthy sentences, this method might offer no gains.

4.2 Semantical Approches

The study by RANASINGHE; ORASAN; MITKOV (2020) represents one of the first attempts to apply neural models to the field of translation memory retrieval. The authors investigated whether the quality of retrieved segments could be improved by using neural models from the Universal Sentence Encoder (USE) collection in the translation memory retrieval pipeline.

The authors evaluated two types of architectures from USE, the Deep Averaging Network (DAN) and a transformer encoder architecture based on the attention mechanism. The DAN architecture averages together word and bi-gram level embeddings and passes them through feedforward layers. The second model is a default Transformer achitecture detailed at 2.3.

To evaluate the models, the authors first tested them on Semantic Textual Similarity

(STS) datasets, specifically SICK and STS2017, and compared the results with edit distance. The DAN architecture performed better on the SICK dataset, while the transformer-based architecture performed better on the STS2017 dataset. In both cases, edit distance showed very low performance for semantic textual similarity. One of the main conclusions is that the neural models handled better when two or more words were changed from the input sentence, but the meaning remained the same.

In the second phase of their study, the researchers aimed to assess the effectiveness of USE models in the retrieval of translation memories. They use an open source DGT-TM, same that we use for our experiments, they set the volume 1 from year 2018 as the index TM and the volume 3 from the year 2018 as the query TM. The TM was indexed using a vector store database AquilaDB.

The authors evaluated the quality of the top 1 retrieved segment using METEOR. They argue that METEOR, due to its use of multistage alignment that includes not only exact matches but also matches words by their base form or synonyms, is a more suitable metric for this type of semantic model analysis. However, we contend that neural models have demonstrated better correlations with human judges (FREITAG et al., 2022), and therefore, should be a better way of evaluating semantic models, as we discuss later.

In the problem of improving machine translation with translation memories, CAI et al. (2021) have utilized monolingual data in the translation generation process. Given a source sentence x and a translation memory, a transformer model trained to generate the target sentence y based on the conditional probability $p(y|x,x_1,...,x_k)$, where $(x_1,...,x_k)$ are target segments most similar to x, that serves as a context for translation. These top k segments are retrieved calculating the dot product of the representations from x and segments stored in the TM. The authors primarily focused on the quality of the translated texts and did not make a deeper study into how this retrieval phase impacts the overall translation quality. A similar approach, but using statistical machine translation that uses paraphrases in the context translation context, can be found in (BIÇICI; DYMETMAN, 2008).

In addition to the aforementioned approach, a Translation Unit (TU) retrieval pipeline has been implemented for in-context learning of new Large Language Models (LLMs) as demonstrated in studies such as HENDY et al. (2023), LIU et al. (2021), and PENG et al. (2023). These studies have utilized sentence encoders like LaBSE to encode sentences from a translation memory and retrieve the top-k results to be used as context for an LLM, such as GPT-3 BROWN et al. (2020), or LAMMA TOUVRON et al. (2023). However, the primary focus of these studies is evaluating the quality of the final translation generated by the LLM models, without directly assessing the impact of TU retrieval models in this scenario. Furthermore, these studies often employ a top-k retrieval approach, with k sometimes reaching as high as 50, but little is done to evaluate how the order and quality of the retrieved segments would impact the final translation.

A work from AGRAWAL et al. (2022) investigated the previously mentioned scenario and proposed an optimized approach for selecting the top-k results in the context of in-context

learning for machine translation tasks. The authors demonstrated that choosing top-k examples that covered all the terms from the input text, yields better translations results. The authors also tested the results against BM25 GARCIA (2009), having a better results in every language direction for the WMT 19 test dataset. While the findings from the study by AGRAWAL et al. (2022) suggest that their proposed ranking approach enhances the performance of machine translation systems, the ranking itself may not be directly applicable for TM purposes. This is because the approach primarily aims to maximize the coverage of words from the input sentence, which might not necessarily return the best matching and most semantically similar sentence from the translation memory.

Traditional machine learning approaches have been investigated for their potential in assessing the similarity between two text segments in the context of Translation Memories. In the study conducted by (GUPTA; BECHARA; ORăSAN, 2014), the authors employed a support vector machine (SVM) to develop a regression model capable of predicting semantic similarity between two text sentences. The features for the SVM model comprised various linguistic features, including surface form, part of speech, lemmas, dependency parsing, paraphrasing, machine translation evaluation, and others. The authors conducted experiments using the DGT-Translation Memory and randomly selected segments for both the training and testing sets. They employed English as the source language and French as the target language for their evaluation. To assess the performance of their approach, they compared it against an edit distance baseline, utilizing the OmegaT software (OmegaT Team, 2024) as a reference for comparison. The results of the experiments consistently showed that the edit distance method, despite its widespread use in open-source Translation Memories software, yielded again relatively poor performance when compared to more semantic approaches.

Cross encoder models, or reranking models, have also gained significant traction within the research community. Notably, the 2019 Deep Learning Track at the Text REtrieval Conference (TREC) (CRASWELL et al., 2020) provided a comprehensive evaluation of the efficacy of BERT model for retrieval tasks. Specifically, on the application of BERT models for reranking coupled with an initial phase that employed the BM25. For Cross-lingual Information Retrieval (CLIR), a problem closely related to Translation Memory Retrieval, the multilingual BERT model was employed by JIANG et al. (2020) to estimate the relevance of queries and documents in different languages. The inputs for the BERT model were a pair of a query q in English and a sentence s in a foreign-language. The output of the $[CLS]$ is then passed through a feed-forward layer to predict the relevance score $p(q|s)$ of the query given the sentence. In the study by Nogueira et al. (2019) on passage re-ranking, a BERT model was fine-tuned for second-stage retrieval. The first stage involved retrieving important documents using BM25, after which the BERT model was employed as a re-ranker. The fine-tuning process utilized a cross-entropy loss function (MAO; MOHRI; ZHONG, 2023) to optimize the model's performance in the re-ranking task.

5

Datasets

This chapter presents and analyzes the key characteristics of the Translation Memories we have used in this work, the DGT-TM, United Nations, Global Voices, and KDE4 Translation Memories. First we introduce and describe each one, following we detail its characteristics, such as the number of Translation Units, percentage of missing segments and segment lengths. The chapter also provides illustrative examples to highlight the unique domains of each dataset.

5.1 Translation Memories

We assessed the efficiency of our approach across a wide spectrum of domains. For that, we utilized the following four public available translation memories for our experiments:

DGT-TM: The DGT Translation Memory is a multilingual resource provided by the European Commission STEINBERGER et al. (2012). It primarily encompasses summaries of EU legislation, including treaties, regulations, and directives adopted by the European Union (EU). It provides translations in the 24 european languages. This Translation Memory is commonly used by researchers for translation memory retrieval experiments RANASINGHE; ORASAN; MITKOV (2020), KIM et al. (2021). We have included it in our Translation Memory list to facilitate method comparison with previous approaches. The DGT-TM, which has been released since 2007, is organized into different volumes by year. For our experiments, we selected Volume 1 and Volume 3 from the year 2018. Translation Units extracted from Volume 1 will be used to construct the index, while those from Volume 3 will comprise the query dataset.

United Nations: The United Nations Parallel Corpus version one is a collection of official records and parliamentary documents sourced from the United Nations ZIEMSKI; JUNCZYS-DOWMUNT; POULIQUEN (2016). These documents have undergone manual translation into the six official UN languages over the years. To align the sentences and paraphrases in these documents, an alignment process was employed using the Bleu-Champ Sentence Alignment method[1]. This method requires a translation model to generate sentences in the same language,

[1] https://github/emjotde/bleu-champ

which are then compared based on matching n-grams to achieve alignment.

The corpus was created mainly to support and provide access to multilingual resources for research in natural language processing tasks, including machine translation. The corpus is available online [2] as bilingual and in a six-language parallel corpus subset.

We divided the TM into yearly segments. For our index TM, we designated the data from 1994, and for our query dataset, we used the data from 1995. To expedite the experimental process, we chose to extract a sample of 10,000 from the 1995 dataset. We believe this sample size does not affect the final results significantly, as it has enough examples to represent the data from 1995.

Global Voices: The Parallel Global Voices is a collection of parallel corpora derived from the Global Voices group websites PROKOPIDIS; PAPAVASSILIOU; PIPERIDIS (2016). Global Voices is a diverse community of volunteers and journalists engaged in writing, summarizing, and translating any event.

The Global Voices TM differs from the DGT-TM and United Nations TM in that it lacks a structured temporal division for creating index TM and query segments. Unlike the first two TMs, Global Voices data is not released by volumes or years. Consequently, performing a train-test split based on time was not feasible. Instead, we chose to randomly select 10,000 translation units for the test dataset and kept the remainder for training.

KDE4: The KDE4 Application collection is openly accessible through the Opus project TIEDEMANN (2012). This corpus encompasses an extensive array of 92 languages and more than 4,000 bilingual text files from the KDE4 localization files, an open source software for desktop and portable computing. Similar to the approach taken with the Global Voices parallel corpus, we applied English pivoting to create the multilingual dataset. The subdivision into training and testing datasets was done through a random sampling procedure, resulting in the allocation of 10,000 samples within the test dataset.

5.2 Size of the Translation Memories

The total number of translation units is presented at Table 5.1 for both the index and query sets for each dataset. While most numbers are similar across datasets, Global Voices notably contains a large number of translation units. However, it is important to note that not all translation units encompass segments from all languages. Both Global Voices and KDE4 exhibit a substantial degree of missing data, as we will explore further in the next section. For the query segments of the United Nations, KDE4, and Global Voices datasets, we sampled 10,000 segments for each.

[2] https://conferences.unite.un.org/UNCorpus

Dataset	Index size	Query size
DGT-TM	145,490	49,634
United Nations	151,480	10,000
KDE4	160,776	10,000
Global Voices	450,197	10,000

Tabela 5.1: Translation Memories splits, for the train and test set.

5.3 Missing Segments

A TM can store *N* different languages, where each TU may have a set of segments in different languages, which may not necessarily be the same set of languages for every TU. We define missing segments as those instances where a TU does not have a segment for a specific language. In Table 5.2, we present a comprehensive overview of the total number of missing segments for each language within our Translation Memories. Initially, we observe that the United Nations TM encompasses all segments for every language, with no missing segments, for this TM. The DGT-TM has a relatively small number of missing segments for each of its languages, with most languages having less than 5% of the total number of segments missing. The Global Voices and KDE4 datasets exhibit a significant proportion of missing segments, particularly noticeable in Global Voices. In the case of Global Voices, where the source query segments are in Portuguese (PT), we observe a 79% rate of missing segments. Implementing methods that leverage and retrieve data beyond solely relying on the source segments index proves highly advantageous in such scenarios.

As evident from the data, English (EN) appears consistently across all cases. This prevalence stems from the common practice of aligning translations with English as the source language, as observed in our approach for both the Global Voices and KDE4 datasets using pivoting techniques.

5.4 Segments Length

The length of segments in both the source and target sentences can significantly influence the performance of neural models. For instance, models such as LaBSE and XLM-R are trained with input lengths limited to 512 tokens. Exceeding this limit may lead to errors or hinder the model's ability to generalize effectively. To analyze any potential effects, we pass the source and target segments through the XLM-R tokenizer and examine the distribution for each TM, as depicted in Figure 5.1, the majority of the data fall below 200 tokens, with DGT-TM and Global Voices exhibiting longer text lengths, while KDE with the overall lower mean across all TMs.

Additionally, Table 5.3 displays the number of examples from each TM where the token count exceeds the 512-token limit. In all instances, only a small number of examples surpass this limit.

Translation Memory	Language	Missing Percentage
DGT-TM	en,es	0%
	pt,fi,et,bg,lv	2%
	de,da,it,cs,hu,cs,pl,sl,sk	3%
	el,fr,sv,hr	4%
	mt	8%
United Nations	en,fr,ar,es,ru,zh	0 %
Global Voices	en	0%
	it	69%
	es	7%
	fr	26%
	pt	79%
	ru	78%
	zh	82%
KDE4	en, de	0%
	fr	8%
	ru	18%
	zh_CN	35%
	pt_BR	3%
	es	5%
	ja	37%

Tabela 5.2: Missing Segments: Number of missing segments per language for each Translation Memory. DGT-TM exhibits a relatively low missing percentage, with the majority below 5%. United Nations shows no missing data for any language. Global Voices, due to its automated alignment, displays a significant number of missing data for almost all languages. KDE4 stands in the middle, although for the English-German pairs (our translation direction), there is no missing data.

Translation Memories	**>512 Tokens**
DGT-TM	73
United Nations	107
Global Voice	2
KDE	108

Tabela 5.3: Number of input instances that have a count greater than 512 tokens.

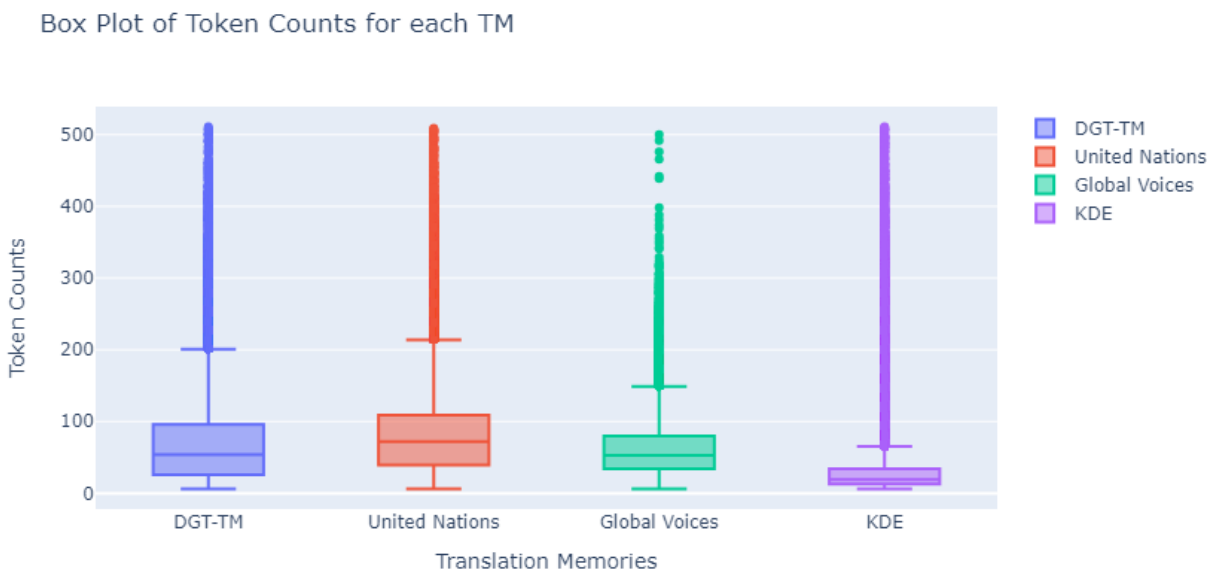

Figura 5.1: Exploring Token Counts: This box plot illustrates the distribution of token counts across input examples for each TM. The majority of data points fall below the maximum 512 token limit for both LaBSE and XLM-R models

5.5 Segment Examples

Below, we provide examples from the Translation Memory, each featuring a collection of five source and target segment pairs sourced and extracted. These examples highlight the distinctions and unique characteristics of each chosen TM.

DGT-TM: For the DGM-T as shown in Table 5.4, we observe that the predominant data pertains to legislation. This ranges from short segments referring to formatting, such as "ANNEX II (A)", section titles like "Decision of the EEA Joint Committee".", and segments from the European Union legislation.

United Nations: The United Nations offers texts similar to those found in the DGT-TM but within the context of United Nations legislation and treaties. Examples are presented in Table 5.5, instances of formatting and titles (examples 1 and 2), along with transcriptions from United Nations meetings.

Global Voices: The Global Voices examples are depicted in Table 5.6. This TM primarily consists of translated news articles or website pages that have been automatically aligned. As evidenced by the examples, the content spans a wide range of subjects but is predominantly sourced from news sites.

KDE4: The KDE4 examples are displayed in Table 5.7. As previously showed, KDE4 TM exhibits relatively short lengths, primarily comprised of small components from the KDE4

Source Segment	Target Segment
ANNEX II (A)	ANEXO II BIS
Decision of the EEA Joint Committee	Decisión del Comité Mixto del EEE
information systems, tools or equipment for sharing information between Member States and third countries,	sistemas, herramientas o equipos de información para el intercambio de información entre los Estados miembros y terceros países,
gal person governed by private law (Article 51(3) of the Rules of Procedure)	presentación del poder si la parte representada es una persona jurídica de Derecho privado (artículo 51, apartado 3, del Reglamento de Procedimiento)
the following point is inserted after point 19ab (Commission Regulation (EU) No 1213/2010):	Después del punto 19ab [Reglamento (UE) n.o 1213/2010 de la Comisión], se inserta el punto siguiente:

Tabela 5.4: DGT-TM Source and Target Segments Examples.

OS interface. This is exemplified by segments in the table as "The passwords are Different"or "Configure the networks". KDE4 also presents examples where the source and target formats differ. For instance, in examples 2 and 4, the source segment does not utilize -"or "&"as seen in the target segment.

Source Segment	Target Segment
Beds a/	Lits a/
PRINCIPLE 28. RESTRICTIONS ON THE PRACTICE OF AMNESTY	PRINCIPE 28 - RESTRICTIONS A LA PRATIQUE DE L'AMNISTIE
The Government, on 28 October 1997, did not deny the arrest and trial of Hendrique Belmiro da Costa, but replied that he was never subjected to torture, and that the source of the allegation was merely exploiting the fact of his very poor health.	Le 28 octobre 1997, le Gouvernement n'a pas nié l'arrestation et le procès de Hendrique Belmiro da Costa mais a répondu qu'il n'avait jamais été torturé et que la source d'information ne faisait qu'exploiter le fait qu'il était en très mauvaise santé.
10. Urges all the Afghan parties to provide efficient and effective remedies to the victims of grave violations of human rights and of accepted humanitarian rules and to bring their perpetrators to trial in accordance with internationally accepted standards;	10. Prie instamment toutes les parties afghanes d'offrir des recours effectifs aux victimes de violations graves des droits de l'homme et des règles humanitaires acceptées et de déférer les auteurs de ces violations aux tribunaux, conformément aux normes internationalement acceptées;
360. The Committee strongly recommends that the Hong Kong Government consider again the adoption of a universal, comprehensive retirement—protection scheme which seeks to ensure that disadvantaged groups are accorded full access to social security.	360. Le Comité recommande vivement au Gouvernement de Hong Kong d'envisager à nouveau d'adopter un système général, d'application universelle, de protection sociale des retraités permettant aux groupes défavorisés d'avoir pleinement accès à la sécurité sociale.

Tabela 5.5: United Nations TM Source and Target Segments Examples.

Source Segment	Target Segment
Praticar yoga e meditação e seguir uma rotina diária de aquietamento e paz são a proposta maior do Yoga pela Paz.	Practicing yoga and meditation and following a daily routine of peace and quieting are the main proposals of Yoga for Peace.
Graffitis estão aparecendo nas paredes de Beirute, sob a forma de placas de sinalização apontando na direção da Palestina.	Graffitti is appearing on the walls of Beirut in the form of signposts pointing in the direction of Palestine.
Por outro lado, Michel, um blogueiro pró-governista, usa o mesmo argumento da Procuradora Geral ao dizer que a liberdade de expressão deve ser equilibrada com a segurança do cidadão e cita a irresponsabilidade de algumas empresas de comunicação [es]:	On the other hand, Michel, a pro-government blogger, uses the same argument as the Attorney General in saying that freedom of expression should be balanced with the citizen security and cites the irresponsibility of some media outlets [es]:
Koffi também publica em seu blog estórias ou críticas produzidas por seus colegas.	Koffi also puts on his blog stories or critiques written by his colleagues.
Você pode experimentar cores, fotos, palavras, etc. O fato que eu comecei a blogar é algo surpreendente; eu nunca achei que pudesse desenvolver um blogue, escrever textos e documentar as coisas novas que estou aprendendo.	You can experiment with colors, photos, words, etc. That I got started with blogging was somewhat surprising; I never thought that I could develop a blog, write texts, and document the new things that I am learning.

Tabela 5.6: Global Voice TM Source and Target Text Examples.

Source Segment	Target Segment
Die PasswÃ¶rter stimmen nicht überein	The passwords are different
UNIX-Erweiterungen	UNIX extensions
In %1 existiert bereits ein Attribut mit dem Namen.	An attribute with that name already exists in %1.
Absteigend sortieren	Sort & Decreasing
Netzwerk einrichten	Configure the network

Tabela 5.7: KDE4 TM Source and Target Segments

6

Robust Translation Memory Retrieval

In this chapter we address the challenge of retrieving more refined and semantically similar segments, particularly when there is no exact match between the source text within the translation memory. For that, we leverage the multilingual nature of Translation Memories. Specifically, we utilize the fact that each translation unit can contain multiple representations of a segment text in different languages. By doing so, we have more context that can be used to increase the likelihood of finding relevant matches within the translation memory.

We take advantage of the latest advances in multilingual encoder models and cross-encoder models and integrate them into the translation memory retrieval pipeline. These models are more capable of capturing the semantic meaning of text, even across different languages and can help identify translation units that are semantically similar to the source text, even if they are not an exact match and are in different languages.

Moreover, we employ Rank Fusion techniques to merge multiple ranks from the TMs. By doing so, we generate a final unified rank that is more robust and exhibits higher quality, enhancing the overall effectiveness of our retrieval process.

In the subsequent sections, we present our proposed solution to the translation memory retrieval problem, namely the RTMR pipeline. We begin by providing an overview of the main components of our pipeline. Then, we delve into each of the components in detail, starting with the neural search and its implementation within our pipeline. We also discuss the use of cross-encoder models and provide details on the training and inference processes. Finally, we describe the used of rank fusion techniques to merge the results of multiple language indexes.

6.1 Robust Translation Memory Retrieval Pipeline

Our pipeline, as depicted in Figure 6.1, consists of two primary steps and an optional one. The first step involves a retrieval process to find the best-matching Translation Unit (TU Retrieval) for a given source segment that a user provides for translation, along with the desired source and target locales. In this step, the user expects to find an exact match, if it exists, or the most semantically similar segment within the TM. To achieve this, first, RTMR employs a neural search across all language indices built from the TM.

For each language, RTMR indexes the segments with a multilingual sentence encoder, which embeds similar segments in different languages in close regions in the embedding space. Then RTMR conducts a similarity search to identify the top k segments that are closest to the source text within each language index.

Once it has identified the most relevant segments for each index, RTMR can proceed to an optional step. This step involves the application of cross-encoder models to re-rank the retrieved segments, basing the new order on their semantic similarity to the source text. This process serves to further enhance the retrieval process, provided that the cross-encoder model can augment each language index beyond the initial neural search.

Finally, in our final step, RTMR employs rank fusion techniques to merge the results of multiple language indexes, here RTMR expects that each rank has its own strength when matching with the source text, and merging them will result in a more stable and robust result. RTMR anticipates that each rank will possess its unique strength when matching with the source text. By merging these ranks, RTMR aims to achieve a more stable and robust outcome. This step is especially beneficial when handling multilingual TMs, as it allows us to harness the strengths of various language models, thereby improving overall retrieval performance.

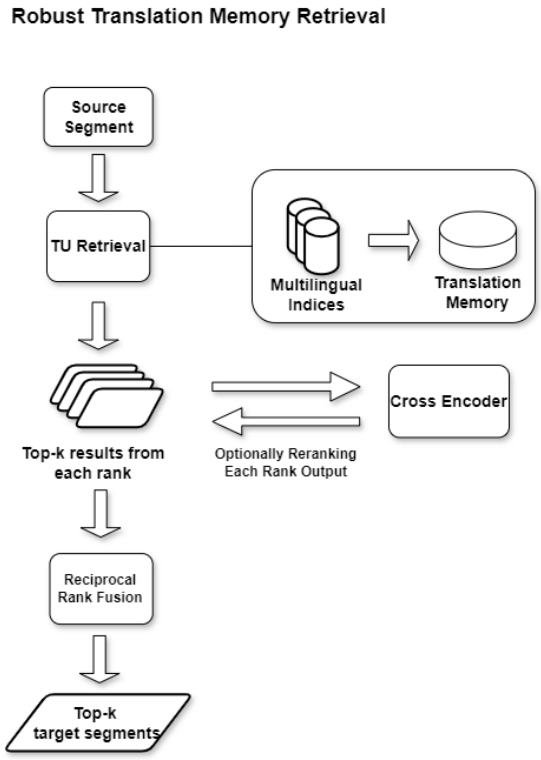

Figura 6.1: TM Retrieval with Reciprocal Rank Fusion: The figure illustrates the query phase for our TM Retrieval solution. Given a new text in language X to be translated to language Y, RTMR generates the input text embedding and for each of our N languages indices RTMR, search for the top k most similar texts. Then RTMR merges the N ranks with Reciprocal Rank Fusion. Optionally each top k results from each rank can be reranked with a Cross Encoder model, then RRF is used to merge the ranks.

6.2 TU Retrieval

The TU Retrieval is essentially a neural search application tailored to address the problem of TM Retrieval within our specific domain. In our context, we deal with TM Retrieval that involves symmetric search and contains multiple representations within each TU. It can be divided into two primary phases: the Indexing phase and the Query phase. Figure 6.2 provides an illustration of both phases, and we will delve into the details of each one below:

Indexing: In this phase, a multilingual bi-encoder model is employed to convert textual data from TUs into dense vectors. The bi-encoder model maps sentences into an embedding space, where semantically similar sentences are close to each other, and dissimilar ones are far distant. These embeddings are subsequently stored using FAISS (JOHNSON; DOUZE; JÉGOU, 2019), an open source library for efficient similarity search and clustering of dense vectors.

Query: In this phase the system receives a text and generates a dense vector with the same multilingual bi-encoder model that transforms the source segment into a dense vector. Then, the dense vector representation of the new segment is compared with the pre-indexed segments stored in the index. The comparison is carried out with the cosine similarity metric 6.1, as our similarity function f from (1.1). The top k segments are returned as the answer.

$$cos(a,b) = \frac{< a.b >}{||a.b||} \tag{6.1}$$

The TU Retrieval is essentially a symmetric search since the indexed texts in the TM and the query texts are relatively similar in content and length, although it can vary a little depending on the source and target languages. As opposed to search engine queries, which is a type of asymmetric search, whereby the query itself is usually small, while the results are typically long texts. This is the main criteria that we used to choose the sentence embedding models.

In our RTMR pipeline, the TU Retrieval stage serves as the initial retrieval phase, responsible for selecting the top k segments that will be further processed in the following stages, wheter the individual ranks are reranked with the optional cross-encoder step or directly merged with rank fusion. The primary goal of this stage could be seen as to filter out less relevant segments and narrow down the search space for the next stages.

6.3 Re-ranking models

The re-ranking stage in our RTMR pipeline is an optional step after the TU Retrieval. This step is considered optional because it can make the RTMR solution more computationally expensive. The purpose of this stage is to refine each language rank before proceeding to the rank fusion step. The idea is that a fine tuned cross-encoder model can refine and better rank the top-k segments retrieved for rank.

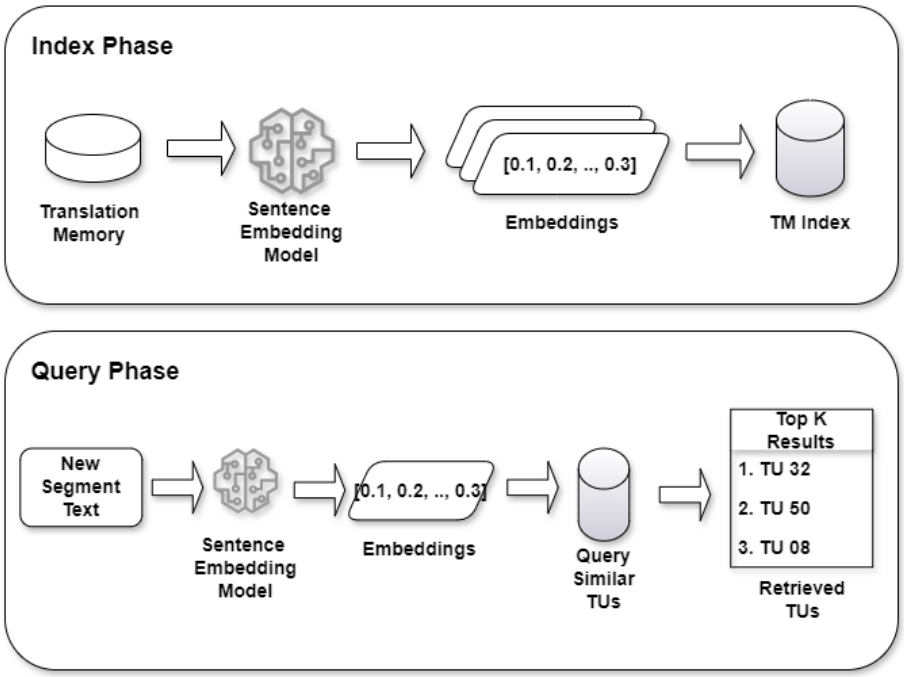

Figura 6.2: TU Retrieval: The figure illustrates the query phase and index phase for TU retreival. During the index phase, all TUs from the TM are indexed. For a selected language, the language segment is passed through a sentence embedding model, and the resulting embeddings are stored in a TM index. In the query phase, a new segment of text to be translated is received and passed through a sentence embedding model. A list of the top k most similar TUs is then returned, based on the similarity of their embeddings to the embedding of the query segment.

The RTMR pipeline leverages the XLM-Roberta (XLM-R) model, a transformer-based multilingual model pre-trained on over 100 languages (CONNEAU et al., 2019). Given a source sentence in a specific language and a target sentence in a different language, the model assesses the relevance between the sentences. It generates a similarity score that can be utilized for re-ranking, effectively treating the task as a Point-Wise Learning to Rank (LTR) problem.

Cross Encoder Architecture The proposed model architecture aligns with the approach presented in section 2.3, where the downstream task is designed to evaluate the relevance score between two sentences, potentially in different languages. The input tokenized format for this model varies slightly from the conventional BERT model. For the XLM-R model the two sentences are demarcated using the $[PAD]$ token, which is also added at the end of the sentence.

First, we tokenize the input sentences, X and Y, using the BERT tokenizer. This process breaks down each sentence into a list of tokens, resulting in two token sequences: $[x_1,\ldots,x_n]$ and $[y_1,\ldots y_p]$. These token sequences are then fed into the XML-R model as follows.

$$\text{input} = [x_1,\ldots,x_n] \text{ and } [y_1,\ldots y_p]$$
$$\text{embeddings} = \text{XML-R}([CLS],x_1,\ldots,x_n,[PAD],y_1,\ldots,y_p,[PAD])$$

Similar to the BERT fine-tuning process, a final language model head is required. The $[CLS]$ token is passed through a feedforward neural network. However, instead of applying a Softmax function, we are only interested in generating a single score value, resulting in a single relevance score output. The final score output can be represented as:

$$\text{output} = W \cdot [\text{CLS}]_{embedding} + b \tag{6.2}$$

Loss Function: We used two rank loss functions for finetuning our model, the first is presented at FORMAL; PIWOWARSKI; CLINCHANT (2021) and showed in equation 6.3.

$$L_{\text{rank}} = -\log \frac{e^{s(q_i,d_i^+)}}{e^{s(q_i,d_i^+)} + e^{s(q_i,d_i^-)}} \tag{6.3}$$

Where:

- s representes our model.
- q_i the source sentence.
- d_i a target sentence, where d_i^+ a target sentence there is a positive example and d_i^- a negative example.

The rank loss function penalizes the model when it assigns a higher score to the negative example compared to the positive one, as example, $s(q_i,d_i^-) > s(q_i,d_i^+)$. However, if $s(q_i,d_i^+) > s(q_i,d_i^-)$ the model weights will not be significantly adjusted because the model correctly ranks the examples, regardless of the relevance magnitude.

The second rank loss is an upgrade of the previsouly one, with the use of in-batch-negative sampling (FORMAL et al., 2021). The use of in-batch-negatives is simply to add negative examples from the training batch. The model also penalizes cases where the ranking is incorrect for the examples within the batch, specifically when $s(q_i,d_i^{INB}) > s(q_i,d_i^+)$.

$$L_{\text{rank}} = -\log \frac{e^{s(q_i,d_i^+)}}{e^{s(q_i,d_i^+)} + e^{s(q_i,d_i^-)} + \sum_j^N e^{s(q_i,d_{i,j}^-)}} \tag{6.4}$$

In this equation:

- N is the size of the batch.
- and $d_{i,j}^-$ is the j negative example from the batch.

Training Data: In the context of Information Retrieval (IR), a common scenario involves a collection of query-document pairs used to train a neural model. Typically, the query is a concise text representing the user's need, while a document can range from a paragraph to an entire document. However, in Translation Memory Retrieval (TMR), our training data is restricted to

the textual content found within translation units. The model should be capable of assessing the relevance of a new input segment to a specific translation unit segment, regardless of the language in which this segment is written.

A source segment q_i, a positive segment d_i^+, and a negative segment d_i^-. We define these as the anchor, positive, and negative segments, respectively.

To construct the training dataset for fine-tuning our XLM-R model, we need to establish a method for selecting positive examples, where the target segment is relevant to the source segment, and negative examples, where the target segment is not relevant.

One issue we encounter is that we cannot construct pairs with the same language. Doing so would make it excessively easy for the model to distinguish positive pairs, as the source and target segments would essentially be identical, and negative pairs would simply consist of different texts. This would result in the model quickly overfitting. To prevent this, we ensure that the pairs always have different languages.

Given a TM, our training dataset is constructed as follows: First, the anchor and positive examples are randomly selected from a TU within the TM. Then, we randomly select two different languages from the TU, and extract their segments. For the negative pairs, we employ two different approaches: Random Sampling and Hard Sampling, which are described below.

For Random Sampling we randomly select a translation unit TU_j from the TM. Then, we randomly select a language from the TU_j and extract its segment. This approach may provide easy examples to the model, as there is a high probability that the source segment will be very different from the target, even in basic lexical aspects, like the lenght of text. To mitigate this, we filter the initial TU set so that only those with similar length, at least 20 characters within, can be selected.

In situations where the model needs to rank closely related sentences, the Random Sampling approach might not be effective, as the resulting negative segment could still be very different from the source segment. To create hard samples, we first apply TU Retrieval to index our Translation Memory (TM). For each anchor segment, we perform a neural search to find the top k most similar segments. The last returned segment, the top k sentence, is used as a hard negative sample. Given that the top k segments are ranked based on their similarity to the anchor segment, from the TU Retrieval, the last returned segment serves as a more difficult negative example, as it is still somewhat similar to the anchor segment but less than the other segments in the top k result.

6.4 Rank Fusion

Previous approaches utilized only a single index for the Translation Memory Retrieval problem (RANASINGHE; ORASAN; MITKOV, 2020), (KIM et al., 2021). In contrast, our solution takes advantage of the multiple translated segments for the same Translation Unit by retrieving the top-k segments of each language index for a given source segment. This strategy

allows our model to benefit from the diverse representations provided by the different languages.

Our solution combines the results from the multiple language indices into a final rank using a rank fusion algorithm. Specifically, we use the Reciprocal Rank Fusion (RRF) algorithm (CORMACK; CLARKE; BUETTCHER, 2009), to return an unified top-k segments for the input source segment. Given a set of TU from the TM, an input segment q, the RRF score for our problem is calculaed as follows:

$$RRF(tu \in TM) = \sum_{r \in R} \frac{1}{k + r(tu)} \tag{6.5}$$

Where:

- R is a list of ranks for each language, returned from the TU Retrieval phase for the q input segment..
- $r(tu)$ a function that returns the TU position for the r rank.

The RRF algorithm favors higher-positioned segments and those segments that appear in multiple ranks.

Since a Translation Memory can have a varying number n of individual language indexes, there are a total of $2^n - (n+1)$ ways of selecting the ranks for merging. In our research, we explored a diverse set of strategies for selecting subsets of indexes and narrowed them down to two different approaches:

- Source and Target: In this method, we exclusively utilized the source and target indexes, taking into account the source sentence language and the desired translation. This approach emerged as the most promising and effective out-of-the-box solution in our investigation.
- Multiple Indexes: This approach involved selecting and indexing all available languages within our translation memory, followed by merging the indices.

7

Experimental Setup

This chapter outlines the experimental setup for our RTMR solution. First, it details the chosen translation directions, cleaning and alignment. Following our evaluation methods and statistical tests for the Translation Memory Retrieval problem. The chapter concludes with a list of models presented in the results chapter, including our own solution, baselines, and related work, along with their respective parameters.

7.1 Translation Directions

The ultimate goal of a TU Retrieval pipeline is to provide the top-k most similar segments in a target language based on a given source segment input. In our experiments, we have the flexibility to select from various combinations of source and target languages. To streamline the process, we opted to fix a specific set of source and target languages for each Translation Memory (TM). The selected language directions are outlined in Table 7.1.

For the DGT-TM, we maintained the English to Spanish direction, a common choice among practitioners in this field (RANASINGHE; ORASAN; MITKOV, 2020), (KIM et al., 2021), to facilitate comparison with existing results. In the case of the United Nations TM, we chose the English to French direction. As for our last two TMs, KDE4 and Global Voices, we deliberately switched the language roles, making English the target language for both. For KDE4, German serves as the source language, while for Global Voices, Portuguese is the source.

By utilizing a diverse set of Translation Memories with distinct translation directions, we aim to demonstrate the robustness of our results across various scenarios.

7.2 Preprocessing and Cleaning

We employ a straightforward pipeline to clean and prepare our Translation Memories for experimentation. Initially, we purged all duplicate Translation Units entries featuring identical source and target texts. This step is crucial for all the methods that depend on Rank Fusion or Reranking. Oftentimes, the returned top-k segments retrieved from an index can be overloaded with repeated segments, compromising the efficacy of such methods. For Rank Fusion, it enhances the score of repeated segments, boosting their importance. As for reranking methods,

Dataset Name	Source	Target
DGT-TM	English	Spanish
United Nations	English	French
KDE4	German	English
Global Voices	Portuguese	English

Tabela 7.1: Translation Memories source and target languages direction.

it decreases the number of unique segments that require ranking, thereby lowering the overall recall.

Secondly, we retained only Translation Units where the target segment is present. Translation Units without the target segment would be useless for TU Retrieval, although we could match the source to similar Translation Units, we wouldn't be able to return a target segment for this Translation Unit.

Finally, the Translation Unit segments were not modified at any stage of the pipeline. We retained both very short and long texts while preserving their case sensitivity. Although lengthy text can impact neural models' output [1], for our TU Retrieval task, we focus on assessing the performance of each approach in real-world scenarios, where the presence of very long segment text can be a reality.

7.3 Evaluation Metrics

In this work, we assess the performance of these systems by examining the overall translation quality of the top 1 retrieved segment from the translation memory. In order to accurately evaluate performance systems, we need to choose one of the many available machine translation metrics, and one that is well-suited to our experimental setup and can effectively capture the differences in scenarios where there is no predefined correct translation stored in the translation memory.

The BLEU score is a widely adopted machine translation metric PAPINENI et al. (2002). It heavily relies on the overlapping n-grams between the target and reference text, which can result in a low BLEU score for text segments that are semantically similar but have a very different vocabulary. To overcome this limitation, another metric called METEOR BANERJEE; LAVIE (2005) has been used for testing translation memory retrieval. METEOR uses stemming and WordNet to go beyond exact matches of the terms, which allows it to better capture semantic similarity. However, in our experiments, we found that METEOR falls short in identifying similar and closer translations. Additionally, its implementation requires a specific tokenizer for each language and a WordNet-like resource for specific languages, which can be a limitation for its use in certain scenarios. It is worth noting that there are different types and versions of these

[1] The number of tokens can be greater than the model supports. This is generally handled with truncating the text, meaning some information might be lost. For encoder models, this means the text might not fully represent the original. In translation evaluation, the target text length may be limited, resulting in inaccurate assessments of the translation quality.

METEOR metrics, and implementation issues have been found in some cases. For example, a bug was reported in the implementation of the METEOR metric in the Natural Language Toolkit (NLTK) library BIRD; KLEIN; LOPER (2009) [2].

Newly developed neural-based learned metrics exhibit stronger correlations with human judgment as we can see in the WMT 2022 Metrics Tasks by FREITAG et al. (2022), unlike traditional overlap metrics such as BLEU, METEOR, and chrF. These new metrics consist of large language models (LLMs) trained on a large quantity of synthetic and accurate translation data. As a result, they have achieved better results when compared to human evaluation. This suggests that these neural-based learned metrics may be more effective in capturing the semantic similarity between the source and target texts, given that they have a better correlation for the problem of evaluating machine translation. Table 7.2 presents a comparison of 13 translation metrics evaluated in the WMT 2022 Metrics Task. Among these metrics, BLEU has one of the worst results, with a very low correlation with human ratings. On the other hand, neural models like COMET-22 and BLEURT-20, which have been used as alternatives to traditional machine translation evaluation metrics, show a better correlation with human judgment.

Metric	**avg rank**
METRICX XXL	1.20
COMET-22	1.32
UNITE	1.86
BLEURT-20	1.91
COMET-20	2.36
MATESE	2.57
COMETKIWI*	2.70
MS-COMET-22	2.84
UNITE-SRC*	3.03
YISI-1	3.27
COMET-QE*	3.33
MATESE-QE*	3.85
MEE4	3.87
BERTSCORE	3.88
MS-COMET-QE-22*	4.06
CHRF	4.70
F101SPBLEU	4.97
HWTSC-TEACHER-SIM*	5.17
BLEU	5.31
REUSE*	6.69

Tabela 7.2: Official ranking of all primary submissions of the WMT22 Metric Task. Low values means better correlation with human evaluation. Source FREITAG et al. (2022).

In our experiments, we chose to evaluate and compare translation retrieval systems using the BLEURT-20 [3] and chrF metrics.

[2]https://github.com/nltk/nltk/issues/2655

[3]The best metric "METRICX XXL"consist of a very large model that would make the evaluation times

In addition to BLEURT-20, we also decided to report a lexical metric for comparison purposes. Since we are running experiments in different languages, we chose chrF as it avoids the need for special tokenizers, and it is a metric with better correlation among the lexical ones, as shown in Table 7.2. By using both neural-based and lexical metrics, we aim to provide a comprehensive evaluation of the performance of the translation retrieval systems under investigation.

7.4 Edit Distance Bins

In our study, we adopt a methodology similar to that of RANASINGHE; ORASAN; MITKOV (2020), where the test dataset is partitioned into five distinct segments based on the translation edit distance between the source text and the closest matching text in the training dataset. This segmentation approach allows us to evaluate the effectiveness of the retrieval process in scenarios where there is limited overlap between the words in the source and target texts.

To calculate these bins, we follow a specific procedure. For each translation unit in the query set, we identify all segments with the same source language. We then compute the minimum edit distance between the source text and each of the identified segments. The edit distance is normalized but the size of the query segment text.

By analyzing the retrieval results across these different bins, we aim to gain a deeper understanding of the impact of using neural models for semantic retrieval. Specifically, we aim to investigate the effectiveness of these models in scenarios where the search segment is significantly different from the pre-translated segments available in the translation memory. By evaluating the retrieval performance in these scenarios, we hope to better understand the strengths and limitations of neural models for semantic retrieval, and identify potential areas for improvement in future research.

7.5 Hypothesis Tests

Some final comparison of our evaluated TM retrieval systems can show rather similar translation score results, for this we employed the Kruskal-Wallis (KRUSKAL; WALLIS, 1952) test to determine whether there were significant differences in the performance of the translation memory systems under investigation. The Kruskal-Wallis test is a nonparametric statistical test that assesses the differences among three. The Kruskal-Wallis test is nonparametric and does not depend on the sampled distribution being normal.

In addition, if the results of the test indicated that there were indeed significant differences among the groups we proceeded to then conducted a pairwise comparison using the Wilcoxon signed-rank test WOOLSON (2007).

---

impossible.

7.6 Evaluated Methods

Below, we provide a comprehensive list of all the models and approaches evaluated in the subsequent chapters, along with their specific setup details.

BM25: The BM25 is a commonly employed function for estimating the relevance of queries and documents in Information Retrieval (IR) (ROBERTSON; ZARAGOZA et al., 2009). In our configuration, we utilized BM25 to estimate the relevance between the new input segment and the source text for each TU.

Efficient Retrieval with WordNet: Following the approach in KIM et al. (2021), as explained in Section 4.1, we conducted tests both with and without the expansion of terms using WordNet. In our results, we refer to the former as Kim-VSM (Kim-Vector Space Model), and the latter as Kim-VSM-WN (Kim-Vector Space Model-WordNet). For the approach utilizing WordNet, we assigned a value of one to the α parameter for the terms in the query expansion.

Multilingual Models: The LaBSE (FENG et al., 2020), Universal Sentence Encoder (CER et al., 2018), and Multilingual E5 Text Embeddings (WANG et al., 2024) are all bi-encoder models, each capable of generating multilingual embeddings used for TU Retrieval, as detailed in section 6.2. In our analysis, we refer to the Universal Sentence Encoder, LaBSE, and Multilingual E5 Text Embeddings as USE, LaBSE, and ME5 Large (ME5-L) [4], respectively. Within the TU Retrieval framework, each of these models functions by comparing the source language texts from the input with those from the translation memories.

Cross Encoder: For the experiments, a Cross Encoder (CE) is utilized, which integrates a TU Retrieval phase with a subsequent re-ranking phase. The model's characteristics and training are detailed in Section 6.3. Initially, the CE receives as input the new segment in a source locale, and a TU Retrieval is conducted as the first step. Subsequently, the CE model is employed for re-ranking using the top-k returned segments.

All CE models were trained with the source and target sentences being in different languages. Therefore, we match the input segment to the target segment stored in the TMs.

Robust Translation Memory Retrieval: We have experimented with various combinations outlined in Section 6, employing Reciprocal Rank Fusion (RRF) to consolidate the retrieved results from each index.

For all cases, the LaBSE model serves as the base model for generating multilingual embeddings in the TU Retrieval phase.

[4]ME5: The ME5 contains different checkpoints, including the base, large, and instruct-large checkpoints. We abbreviate it as ME5-L for the best performing variant, ME5 Large.

- RRF with Source and Target: This involves conducting TU Retrieval on both the source and target indexes, followed by merging the results using RRF.
- RRF with Multiple Indices: Here, TU Retrieval is performed across multiple language indexes, and the results are merged using RRF.
- RRF + Cross Encoder: In the initial phase, TU Retrieval is executed separately for both the source and target indexes. Subsequently, each index undergoes individual re-ranking using a cross encoder model before being merged using RRF.

8

Results

In this chapter, we present the results for both our proposed RTMR approaches and the baseline methods. We begin by showcasing the primary outcomes of our experiments, demonstrating how our RTMR pipeline enhances translation retrieval quality across various TMs. We discuss the implications of employing Cross Encoder models in our approach and evaluate the performance of lexical models. Additionally, we conduct an analysis of translation memory retrieval performance based on different edit distance bins and provide an overview of the overall performance of neural models and the gains obtained by employing our RTMR approach.

8.1 RTMR Results

Table 8.1 presents the chrF and BLUERT scores across all approaches. RTMR consistently outperformed other approaches in terms of BLUERT scores. Specifically, RRF with Src/Tgt (ME5-L) yielded the highest BLUERT scores for DGT-TM (0.344) and United Nations (0.351) datasets, RRF with Multiple Indices for Global Voices (0.334), and RRF + CE Src/Tgt for KDE4 (0.450).

DGT-TM: When comparing RRF with Src/Tgt (ME5-L) to the baseline ME5 Large, while there was no improvement in BLEURT scores, chrF scores increased from 0.330 to 0.338. Additionally, RRF with Src/Tgt showed improvement over the LaBSE baseline, achieving a BLUERT score of 0.34 and the best chrF score of 0.351.

United Nations: RRF with Src/Tgt (ME5-L) achieved the highest BLUERT score for the United Nations dataset. Both RRF with Src/Tgt and RRF + CE Src/Tgt showed improvement over the LaBSE baseline. Notably, USE achieved the best chrF score for this dataset.

Global Voices: In contrast to the others TMs, the RRF with Many Indexes obtained the best BLUERT result (0.334), with a big gap against the second one RRF + CE Src/Tgt (0.314). This performance can be attributed to the segment distribution in the TM, particularly with over 80% of missing data for the source locale pt (refer to Section 5.3). The utilization of Many Indices

Dataset	Languages	Model	chrF	BLEURT
DGT-TM	en → es	BM25	0.347	0.31
		Kim-VSM	0.311	0.291
		Kim-VSM-WN	0.296	0.256
		USE	0.347	0.324
		LaBSE	0.349	0.337
		ME5 Large	0.330	**0.344**
		Cross Encoder	0.345	0.332
		RRF with Src/Tgt	**0.351**	0.340
		RRF with Src/Tgt (ME5-L)	0.338	**0.344**
		RRF with Many Indexes	0.345	0.332
		RRF + CE Src/Tgt	0.34	0.337
United Nations	en → fr	BM25	0.407	0.313
		Kim-VSM	0.372	0.299
		Kim-VSM-WN	0.368	0.273
		USE	**0.409**	0.328
		LaBSE	0.399	0.331
		ME5 Large	0.401	0.350
		Cross Encoder	0.382	0.329
		RRF with Src/Tgt	0.405	0.335
		RRF with Src/Tgt (ME5-L)	0.403	**0.351**
		RRF with Many Indexes	0.401	0.331
		RRF + CE Src/Tgt	0.390	0.334
Global Voices	pt → en	BM25	0.217	0.269
		Kim-VSM	0.187	0.261
		Kim-VSM-WN	0.156	0.223
		USE	0.221	0.297
		LaBSE	0.222	0.304
		ME5 Large	0.213	0.301
		Cross Encoder	**0.278**	0.31
		RRF with Src/Tgt	0.224	0.307
		RRF with Src/Tgt (ME5-L)	0.208	0.303
		RRF with Many Indexes	0.242	**0.334**
		RRF + CE Src/Tgt	0.228	0.314
KDE4	de → en	BM25	0.241	0.336
		Kim-VSM	0.231	0.320
		Kim-VSM-WN	0.202	0.281
		USE	0.266	0.392
		LaBSE	0.281	0.421
		ME5 Large	0.279	0.413
		Cross Encoder	0.281	0.432
		RRF with Src/Tgt	0.291	0.445
		RRF with Src/Tgt (ME5-L)	0.280	0.428
		RRF with Many Indexes	0.259	0.414
		RRF + CE Src/Tgt	**0.296**	**0.450**

Tabela 8.1: Comparing translation quality across TMs using BM25, Kim-VSM-*, USE, LaBSE and ME5-L baselines. Our approach employs two indexing approaches: source and target (Src/Tgt), and Many Indexes, both applying Reciprocal Rank Fusion (RRF) to merge the indexes at the end. We also show the results for the Cross Encoder (CE) models for reranking source index. Additionally, we demonstrate CE with RRF, where both source and target outputs from the LaBSE model are reranked, followed by a final merging of indexes using RRF (CE Src/Tgt + RRF).

likely led to a significant increase in recall for the English target. Despite this, all other RRF approaches demonstrated overall better results compared to the baselines.

KDE4: The top BLEURT scores for KDE4 were achieved by RRF + CE Src/Tgt (0.450) and RRF Src/Tgt (0.445), outperforming TU Retrieval with LaBSE (0.421). RRF with Src/Tgt (ME5-L) also showcased improvement over its single index counterpart ME5 Large, with BLEURT scores of 0.428 and 0.413, respectively.

Results confirm that our strategy of using multilingual indices combined with Rank Fusion is effective for translation memory retrieval. A possible reason for that is that performing this task using multiple languages present in the translation memory makes the solution more robust than the ones based on the source language since it might better deal with potential noise in the source index. The results also further confirm the superior ability of neural models to identify semantically similar segments from the TM compared to lexical approaches like BM25 and Kim-VSM-WN.

To emphasize the efficiency of applying RRF to merge the source and target ranks from a TM, Figure 8.1 presents a boxplot comparing BLUERT scores between RRF with Src/Tgt (where LaBSE is used for the TU Retrieval phase) and LaBSE. In this analysis, we excluded cases where the two models produced the same top-1 output, as these instances are likely to represent straightforward or easy cases where both models agree. As it is shown in the plot RRF has a better BLUERT score across all quartiles on the four TM. This indicates the effectiveness of RRF in improving translation quality compared to relying solely on LaBSE.

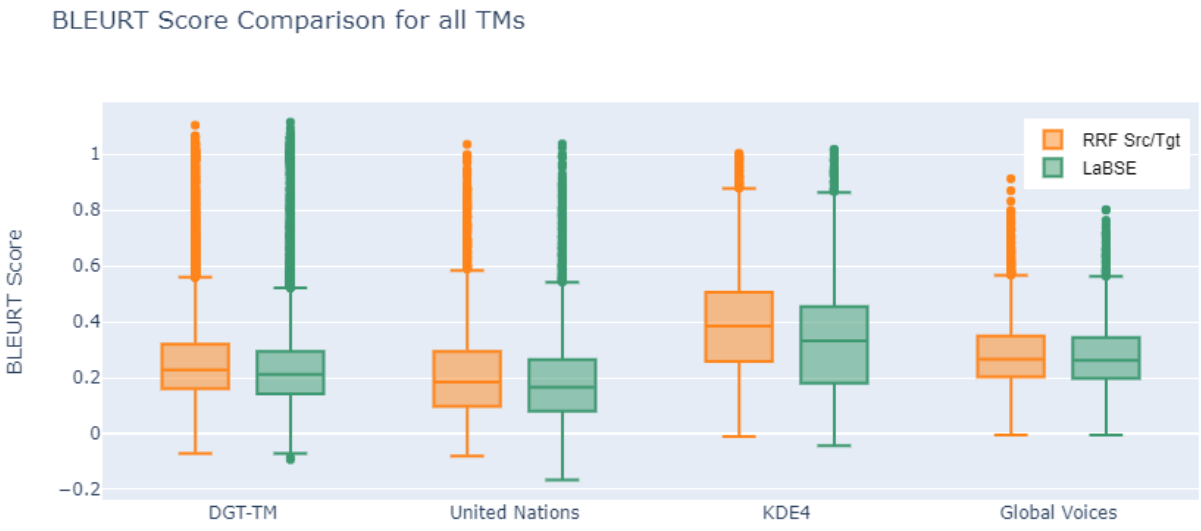

Figura 8.1: RRF Src/Tgt vs Labse boxplot: Applying RRF improved the top-1 TU retrieval performance for all TMs.

Statistical Tests To ensure the validity of our analysis, we conducted statistical tests to determine if the differences between the base model (LaBSE and ME5-L) and the RTMR were

statistically significant. First, we applied the Kruskal test to check for any differences in the groups, and then we validated pairwise comparisons between the base model and RRF.

Table 8.2 presents the results of the Kruskal test. For the DGT-TM, United Nations, and KDE4 datasets, we can conclude that the groups come from different distributions. However, for the Global Voices dataset, the null hypothesis is not rejected. This can be attributed to the fact that ME5-L and ME5-L with RTMT did not show statistically significant differences in the subsequent Wilcoxon tests.

Method	Test Statistic	P-value	Conclusion
DGT	295	8.6e-64	Reject the null hypothesis.
United Nations	317	1.4e-68	Reject the null hypothesis.
Global Voices	0.57	0.44	Fail to reject the null hypothesis.
KDE	106	6.1e-23	Reject the null hypothesis.

Tabela 8.2: Results of the Kruskal-Wallis test for different methods

Table 8.3 presents the pairwise results of the Wilcoxon test, where we compare the effect of the RRF approach with the LaBSE and ME5-Large models. The purpose of this test is to determine if there is a statistically significant difference between the performance of the RRF approach and the two baseline models. Results indicate that using RRF significantly improved the performance of the LaBSE model for all TMs. However, for the ME5-Large model, the RRF approach only showed statistically significant improvements for the DGT-TM and KDE4 datasets.

Dataset	**Model**	**Wilcoxon statistic**	**P-value**	**Significance**
DGT-TM	Labse	558867463.5	4.73e-32	Statistically significant
DGT-TM	ME5 Large	324413.5	7.98e-08	Statistically significant
UN	Labse	2534248.5	1.16e-16	Statistically significant
UN	ME5 Large	23770780.5	0.95	Not statistically significant
Global	Labse	5617639.5	0.003	Statistically significant
Global	ME5 Large	24303988.5	0.47	Not statistically significant
KDE4	Labse	6574163.0	1.12e-60	Statistically significant
KDE4	ME5 Large	22669614.5	2.12e-17	Statistically significant

Tabela 8.3: Comparison of Combine With Labse and ME5 Large Model, ordered by dataset

Impact of Cross Encoder Models The results presented in Table 8.1 demonstrate an improvement over the performance by reranking the source index with a fine tuned cross encoder model, particularly evident in the KDE4 and Global Voices TMs. However, for the DGT-TM and United Nations TM the performance is worse than the baseline model. We believe this discrepancy can be attributed to the varying degrees of curation and cleanliness within these TMs.

The KDE4 and Global Voices TMs exhibit a less curated nature compared to the meticulously curated DGT-TM and United Nations TMs. Notably, the Global Voices TM contains

numerous instances where texts in different languages strict literal translations but rather convey information on the same topic or news. Similarly, within the KDE4 TM, variations exist where the structure of the source text differs from that of the target text.

While the performance for the DGT and United Nations Translation Memories appear to be negatively impacted by reranking, the use of RRF improves its results. Moreover, when RRF is employed on the output of reranked cross encoder models for the KDE4 and Global Voices TMs, where the reranking had a positive effect, a noticeable performance enhancement is also observed. This shows that even for a low or a high quality ranking of the individual indices, applying RRF with Src/Tgt tends to improve the performance. The overall gains can be seen in the following Table 8.4.

TM	BLUERT
DGT-TM	+0.005
United Nations	+0.005
KDE4	+0.018
Global Voices	+0.004

Tabela 8.4: Cross Encoder with source and target indices merged with RRF. The table shows the improvements over the re-raking only the source index.

8.2 Edit Distance Bin Performance

Similar to RANASINGHE; ORASAN; MITKOV (2020), to provide insights about the models' performance regarding how similar is the source segment to its correspondent TU's segment in the same language. For that, we present in Table 8.5, the mean BLEURT scores for each TM divided into five edit distance chunks. The results are presented as $1 -$ Normalized Edit Distance, so lower values suggest scenarios whereby a closely related text may not be available in the TM, while higher values indicate the presence of more similar matching text.

Regarding the results of the DGT-TM dataset, comparing RRF with Src/Tgt (ME5-L) and ME5 Large, both approaches yielded very similar results, with the former showing slight improvements in the first and last bins. However, RRF with Src/Tgt consistently outperformed LaBSE across all bins, with the best results in the (0.2-0.0) bin for all methods.

For the United Nations TM RTMR had better results in the initial (1.0-0.8), (0.8-0.6), and (0.6-0.4) bins, indicating that a higher semantic match for when there is not a similar enoug segment in the TM. The Cross Encoder baseline had the better result for the (0.4-0.2) bins and the LaBSE for the (0.2-0.0) bin.

RRF with Multiple indices had the best results except for the (0.4-0.2) bin for the The Global Voices TM. With the RRF with Src/Tgt having the best result for this bin. As discussed before, these results are greatly influenced by the TM distribution.

Dataset	Model Name	1 - Normalized Edit distance				
		0.0-0.2	0.2 - 0.4	0.4 - 0.6	0.6 - 0.8	0.8-1.0
DGT-TM	BM25	0.171	0.213	0.25	0.367	0.754
	Kim-VSM	0.145	0.197	0.232	0.355	0.746
	Kim-VSM-WN	0.127	0.168	0.195	0.313	0.703
	USE	0.217	0.222	0.25	0.389	0.794
	LaBSE	0.24	0.236	0.265	0.414	0.801
	ME5 Large	0.250	**0.246**	**0.279**	**0.418**	0.800
	Cross Encoder	0.208	0.231	0.259	0.376	0.765
	RRF with Src/Tgt	0.246	0.239	0.268	0.419	**0.803**
	RRF with Src/Tgt (ME5-L)	**0.253**	**0.246**	**0.279**	**0.418**	0.801
United Nations	BM25	0.496	0.276	0.657	0.736	0.852
	Kim-VSM	0.482	0.261	0.681	0.692	0.683
	Kim-VSM-WN	0.433	0.237	0.665	0.598	0.633
	USE	0.553	0.286	0.708	0.746	0.902
	LaBSE	0.576	0.287	0.712	0.746	**0.906**
	ME5 Large	0.573	0.309	0.716	0.743	0.896
	Cross Encoder	0.566	0.286	0.708	**0.759**	0.904
	RRF with Src/Tgt	**0.582**	**0.292**	0.714	0.747	0.903
	RRF with Src/Tgt (ME5-L)	0.582	0.309	0.722	0.751	0.899
	RRF with Multiple indices	0.576	0.288	0.707	0.72	0.901
	CE Src/Tgt + RRF	0.55	**0.292**	**0.722**	0.733	0.897
Global Voices	BM25	0.237	0.26	0.295	0.345	0.355
	Kim-VSM	0.283	0.255	0.344	0.461	0.540
	Kim-VSM-WN	0.250	0.216	0.318	0.431	0.471
	USE	0.333	0.291	0.383	0.449	0.531
	LaBSE	0.359	0.296	0.401	0.523	0.524
	ME5 Large	0.354	0.293	0.403	0.489	0.529
	Cross Encoder	0.344	0.303	0.397	0.475	0.523
	RRF with Src/Tgt	0.367	0.297	0.271	**0.54**	0.52
	RRF with Src/Tgt (ME5-L)	0.361	0.294	0.412	0.509	0.536
	RRF with Multiple indices	**0.393**	**0.325**	**0.438**	0.47	**0.623**
	CE Src/Tgt + RRF	0.365	0.305	0.418	0.529	0.529
KDE4	BM25	0.227	0.321	0.367	0.435	0.537
	Kim-VSM	0.229	0.300	0.348	0.408	0.502
	Kim-VSM-WN	0.207	0.251	0.296	0.362	0.482
	USE	0.313	0.367	0.41	0.48	0.586
	LaBSE	0.35	0.395	0.435	0.354	0.49
	ME5 Large	0.348	0.381	0.430	0.489	0.597
	Cross Encoder	0.373	0.403	0.446	0.511	0.573
	RRF with Src/Tgt	0.375	0.412	0.459	**0.535**	**0.616**
	RRF with Src/Tgt (ME5-L)	0.348	0.397	0.451	0.516	0.612
	RRF with Multiple indices	0.339	0.393	0.437	0.495	0.556
	CE Src/Tgt + RRF	**0.386**	**0.418**	**0.465**	0.53	0.612

Tabela 8.5: Comprehensive BLEURT Score Analysis across Edit Distance Bins: Our approach has superior performance across all edit distance ranges. Including the unexpected 0.8-1.0 range for the Global Voice and KDE4, where we would expect that traditional retrieval methods like BM25 would typically have a better performance.

Finally, for the KDE4, for high values of edit distance (0.6-0.8 and 0.8-1.0), RRF with Src/Tgt achieved the best results, whereas CE Src/Tgt + RRF the best for low values of edit distance. This indicates that the Cross Encoder strategy better captures the semantic similarity since it worked better for less similar segments.

The results suggest that even in scenarios where sentences are highly similar with low edit distances, neural models significantly outperform lexical approaches. Additionally, our RTMR approach exhibited performance improvements across all bins in the United Nations TM, except for the (0.6-0.8) and (0.8-1.0) bins, indicating enhanced accuracy across various cases.

8.3 Output Comparison

In Table 8.6, we highlight three examples that help explain why our approach performs well. The table includes the original text, the desired text, and the results from both the USE and our approach, RRF with Src/Tgt.

The first two examples show instances where the expected translated text is very different from the source text format, suggesting that the TM can have a low quality of TUs or simply that the translation to specify target language in this domain follows a different format.

As one can see, RRF with Src/Tgt , which involves comparing not only the source but also target index, retrieves from the TM the best matches to the translated target text. LaBSE successfully retrieves the best text for the first example but fails in the second one, whereas USE fails in both. The third example underscores a case in which both USE and LaBSE models were able to retrieve quite semantically similar text to the source text. However, our approach ultimately excelled in identifying the best-matching text. Finally, the fourth example presents a situation whereby neither approach retrieved the target text. This could be due to either the absence of such text in the TM or those approaches could retrieve this segment.

Source Text	**Target Text**	**USE**	**LaBSE**	**RRF with Src/Tgt**
"Gewehrschuss@item: inlistbox"	"Gunshot"	"Violin"	"Gunshot"	"Gunshot"
"Telefon@item: inlistbox"	"Telephone"	"Timpani"	Email 2	"Telephone:"
"Schneidet den ausgew0̆0e4hlten Abschnitt aus und kopiert ihn in die Zwischenablage."	"Cuts the selected section and puts it to the clipboard"	"The selected sentences are deleted and placed on the clipboard."	The selected sentences are deleted and placed on the clipboard."	"Cuts the selected section and puts it to the clipboard"
Für die angegebenen Media-Daten kann kein Demultiplexer-Modul gefunden werden.	Cannot find demultiplexer plugin for the given media data	Unable to find a Multimedia Backend	Could not retrieve multi session information from disk.	Unable to find the requested Multimedia Backend

Tabela 8.6: Output Examples: Above are illustrative examples showcasing the superiority performance of our approach compared to previous methods. Notably, RRF with Src/Tgt exhibits robustness in handling noise, as evidenced in the first example, and returns better segments, even when the baseline models provide satisfactory answers, as exemplified in the third row.

9

Conclusions and Future Work

In this chapter, we provide a comprehensive summary of our proposed solution for the Translation Memory Retrieval problem, along with the results obtained from our experiments. We recapitulate our contributions to the field and discuss the potential limitations of our approach. Furthermore, we outline promising avenues for future research to address these limitations and enhance the overall effectiveness of Translation Memory Retrieval systems.

9.1 Conclusion

In this work, we introduced the Robust Translation Memory Retrieval (RTMR) pipeline. Our solution integrates deep neural network models into both the first and second retrieval stages, neural search and cross encoder respectively . Leveraging the multilingual nature of Translation Memory (TM), RTMR allows the creation of multiple language indices (TU). These indices enable the search for similar segments, and Reciprocal Rank Fusion is employed to merge and return the final rank, enhancing the robustness of the retrieval process. An optional step can be run with a cross encoder model, to futher improve the individual language indices before Rank Fusion.

Our analysis demonstrates that the RTMR pipeline, which was tested across four TMs, enhances the quality of returned segments across all scenarios. This improvement is evident both when there are similar segments with overlapping vocabulary and when there are no stored segments similar to the input segment. Futhermore, our approach showed to be more robust to domain specific and quality issues in the TM, where using multiple information improved the search.

9.2 Main Contributions

The main contributions of this work are outlined below:

- Introduction of a Robust Translation Memory Retrieval pipeline integrating neural models, neural search, cross encoders and rank fusion techniques.

- The combination of multiple indices from the TM using Rank Fusion techniques to improve retrieval performance.
- An extensive evaluation of baseline approaches, such as BM25, along with a wide range of neural models that had not been previously assessed in research for TM Retrieval.
- The first evaluation of the use of Cross Encoders for TM Retrieval, along with an analysis of their limitations.

9.3 Potential Limitations

Limitations of this work can include include:

- The applied Rank Fusion techniques assign equal weight to each index, which may not be optimal. Further research is needed to assess the importance of each language index and fine-tune their weights accordingly.
- The proposed pipeline does not take into account efficient retrieval techniques, related to their impact on performance and cost.
- The evaluation was performed on a particular set of language directions, demonstrating improvements over previous research. However, it is important to note that English was utilized as either the source or target language in all cases. To comprehensively assess the pipeline's effectiveness and generalizability, it is necessary to conduct further testing on a more diverse range of language pairs that do not necessarily involve English.

9.4 Future Work

Potential directions and suggestions for future research:

- Development and implementation of a pretrained Cross Encoder specifically for TM Retrieval. This encoder should be trained on an extensive dataset to ensure its ability to generalize effectively for unseen TM domains.
- Exploration and analysis of alternative methods to fine-tune first stage models, such as LaBSE, for specific domains.
- Investigation of the impact of Rank Fusion techniques in scenarios where the initial stage retrieval is conducted via an approximated search. This could potentially enhance the efficiency of the retrieval process.

- Assessment of the influence of RTMR on few-shot approaches that leverage Large Language Models.
- Evaluate the efficiency of the RTMR pipeline for languages that do not involve English as source or target segments.

References

AGRAWAL, S. et al. In-context examples selection for machine translation. **arXiv preprint arXiv:2212.02437**, [S.l.], 2022.

ANAND, M. et al. Serverless BM25 Search and BERT Reranking. In: DESIRES. **Anais...** [S.l.: s.n.], 2021. p.3–9.

ARTHERN, P. J. Machine translation and computerised terminology systems - a translator's viewpoint. In: TRANSLATING AND THE COMPUTER, London, UK. **Anais...** Aslib Proceedings, 1978.

BAEZA-YATES, R. Modern Information Retrieval. **Addison Wesley google schola**, [S.l.], v.2, p.127–136, 1999.

BANERJEE, S.; LAVIE, A. METEOR: an automatic metric for mt evaluation with improved correlation with human judgments. In: OF THE ACL WORKSHOP ON INTRINSIC AND EXTRINSIC EVALUATION MEASURES FOR MACHINE TRANSLATION AND/OR SUMMARIZATION. **Proceedings...** [S.l.: s.n.], 2005. p.65–72.

BARRAULT, L. et al. SeamlessM4T-Massively Multilingual & Multimodal Machine Translation. **arXiv preprint arXiv:2308.11596**, [S.l.], 2023.

BasicCat. **BasicCat - An open source and free computer-aided translation tool**. 2024.

BERGSTRA, J. et al. Algorithms for hyper-parameter optimization. **Advances in neural information processing systems**, [S.l.], v.24, 2011.

BIÇICI, E.; DYMETMAN, M. Dynamic translation memory: using statistical machine translation to improve translation memory fuzzy matches. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TEXT PROCESSING AND COMPUTATIONAL LINGUISTICS. **Anais...** [S.l.: s.n.], 2008. p.454–465.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural language processing with Python**: analyzing text with the natural language toolkit. [S.l.]: "O'Reilly Media, Inc.", 2009.

BLOODGOOD, M.; STRAUSS, B. Translation memory retrieval methods. **arXiv preprint arXiv:1505.05841**, [S.l.], 2015.

BOMMASANI, R. et al. On the opportunities and risks of foundation models. **arXiv preprint arXiv:2108.07258**, [S.l.], 2021.

BROWN, T. et al. Language models are few-shot learners. **Advances in neural information processing systems**, [S.l.], v.33, p.1877–1901, 2020.

CAI, D. et al. Neural machine translation with monolingual translation memory. **arXiv preprint arXiv:2105.11269**, [S.l.], 2021.

CER, D. et al. Universal sentence encoder. **arXiv preprint arXiv:1803.11175**, [S.l.], 2018.

CHEN, B.; CHERRY, C. A systematic comparison of smoothing techniques for sentence-level BLEU. In: OF THE NINTH WORKSHOP ON STATISTICAL MACHINE TRANSLATION. **Proceedings...** [S.l.: s.n.], 2014. p.362–367.

CHUNG, H. W. et al. Rethinking embedding coupling in pre-trained language models. **arXiv preprint arXiv:2010.12821**, [S.l.], 2020.

CONNEAU, A. et al. Unsupervised cross-lingual representation learning at scale. **arXiv preprint arXiv:1911.02116**, [S.l.], 2019.

CORMACK, G. V.; CLARKE, C. L.; BUETTCHER, S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 32. **Proceedings...** [S.l.: s.n.], 2009. p.758–759.

COSTA-JUSSÀ, M. R. et al. No language left behind: scaling human-centered machine translation. **arXiv preprint arXiv:2207.04672**, [S.l.], 2022.

CRASWELL, N. et al. Overview of the TREC 2019 deep learning track. **arXiv preprint arXiv:2003.07820**, [S.l.], 2020.

DEVLIN, J. et al. Bert: pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, [S.l.], 2018.

FELLBAUM, C. **WordNet**: an electronic lexical database. [S.l.]: Bradford Books, 1998.

FENG, F. et al. Language-agnostic BERT sentence embedding. **arXiv preprint arXiv:2007.01852**, [S.l.], 2020.

FORMAL, T. et al. SPLADE v2: sparse lexical and expansion model for information retrieval. **arXiv preprint arXiv:2109.10086**, [S.l.], 2021.

FORMAL, T.; PIWOWARSKI, B.; CLINCHANT, S. SPLADE: sparse lexical and expansion model for first stage ranking. In: INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 44. **Proceedings...** [S.l.: s.n.], 2021. p.2288–2292.

FOX, E.; SHAW, J. Combination of multiple searches. **NIST special publication SP**, [S.l.], p.243–243, 1994.

FREITAG, M. et al. Results of WMT22 Metrics Shared Task: stop using bleu – neural metrics are better and more robust. In: SEVENTH CONFERENCE ON MACHINE TRANSLATION, Abu Dhabi. **Proceedings...** Association for Computational Linguistics, 2022. p.46–68.

GALA Global. **TMX 1.4b Specification**. 2024.

GANITKEVITCH, J.; VAN DURME, B.; CALLISON-BURCH, C. PPDB: the paraphrase database. In: HUMAN LANGUAGE TECHNOLOGIES, 2013. **Proceedings...** [S.l.: s.n.], 2013. p.758–764.

GARCIA, I. Beyond translation memory: computers and the professional translator. **The Journal of Specialised Translation**, [S.l.], v.12, n.12, p.199–214, 2009.

GARG, A.; AGARWAL, M. Machine translation: a literature review. **arXiv preprint arXiv:1901.01122**, [S.l.], 2018.

GUO, M. et al. Effective Parallel Corpus Mining using Bilingual Sentence Embeddings. In: THIRD CONFERENCE ON MACHINE TRANSLATION: RESEARCH PAPERS, Brussels, Belgium. **Proceedings...** Association for Computational Linguistics, 2018. p.165–176.

GUPTA, R.; BECHARA, H.; ORăSAN, C. Intelligent translation memory matching and retrieval metric exploiting linguistic technology. In: TRANSLATING AND THE COMPUTER 36. **Proceedings...** [S.l.: s.n.], 2014.

GUPTA, R. et al. Improving translation memory matching and retrieval using paraphrases. **Machine translation**, [S.l.], v.30, p.19–40, 2016.

HAN, S. et al. Learning-to-Rank with BERT in TF-Ranking. **arXiv preprint arXiv:2004.08476**, [S.l.], 2020.

HENDY, A. et al. How good are gpt models at machine translation? a comprehensive evaluation. **arXiv preprint arXiv:2302.09210**, [S.l.], 2023.

JAISWAL, A. et al. A survey on contrastive self-supervised learning. **Technologies**, [S.l.], v.9, n.1, p.2, 2020.

JIANG, Z. et al. Cross-lingual information retrieval with BERT. **arXiv preprint arXiv:2004.13005**, [S.l.], 2020.

JOHNSON, J.; DOUZE, M.; JÉGOU, H. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, [S.l.], v.7, n.3, p.535–547, 2019.

KIM, K.-h. et al. Translation Memory Retrieval Using Lucene. In: INTERNATIONAL CONFERENCE ON RECENT ADVANCES IN NATURAL LANGUAGE PROCESSING (RANLP 2021), Held Online. **Proceedings...** INCOMA Ltd., 2021. p.684–691.

KINGMA, D. P.; BA, J. Adam: a method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, [S.l.], 2014.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American statistical Association**, [S.l.], v.47, n.260, p.583–621, 1952.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, [S.l.], v.521, n.7553, p.436–444, 2015.

LEE, S. et al. A survey on evaluation metrics for machine translation. **Mathematics**, [S.l.], v.11, n.4, p.1006, 2023.

LIU, J. et al. What Makes Good In-Context Examples for GPT-3? **arXiv preprint arXiv:2101.06804**, [S.l.], 2021.

LIU, T.-Y. et al. Learning to rank for information retrieval. **Foundations and Trends® in Information Retrieval**, [S.l.], v.3, n.3, p.225–331, 2009.

LOPEZ, A. Statistical machine translation. **ACM Computing Surveys (CSUR)**, [S.l.], v.40, n.3, p.1–49, 2008.

LU, Y. et al. Ernie-search: bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. **arXiv preprint arXiv:2205.09153**, [S.l.], 2022.

MAO, A.; MOHRI, M.; ZHONG, Y. Cross-entropy loss functions: theoretical analysis and applications. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING. **Anais...** [S.l.: s.n.], 2023. p.23803–23828.

MITRA, B.; CRASWELL, N. et al. An introduction to neural information retrieval. **Foundations and Trends® in Information Retrieval**, [S.l.], v.13, n.1, p.1–126, 2018.

OmegaT Team. **OmegaT - The Free Translation Memory Tool**. 2024.

ÖSTLING, R. et al. The Helsinki neural machine translation system. **arXiv preprint arXiv:1708.05942**, [S.l.], 2017.

PAPINENI, K. et al. Bleu: a method for automatic evaluation of machine translation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 40. **Proceedings...** [S.l.: s.n.], 2002. p.311–318.

PENG, K. et al. Towards making the most of chatgpt for machine translation. **arXiv preprint arXiv:2303.13780**, [S.l.], 2023.

POPOVIĆ, M. chrF: character n-gram f-score for automatic mt evaluation. In: OF THE TENTH WORKSHOP ON STATISTICAL MACHINE TRANSLATION. **Proceedings...** [S.l.: s.n.], 2015. p.392–395.

PROKOPIDIS, P.; PAPAVASSILIOU, V.; PIPERIDIS, S. Parallel Global Voices: a collection of multilingual corpora with citizen media stories. In: TENTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2016), Paris, France. **Proceedings...** European Language Resources Association (ELRA), 2016.

PU, A. et al. Learning compact metrics for MT. **arXiv preprint arXiv:2110.06341**, [S.l.], 2021.

RANASINGHE, T.; ORASAN, C.; MITKOV, R. Intelligent Translation Memory Matching and Retrieval with Sentence Encoders. In: ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR MACHINE TRANSLATION, 22., Lisboa, Portugal. **Proceedings...** European Association for Machine Translation, 2020. p.175–184.

REIMERS, N.; GUREVYCH, I. Sentence-BERT: sentence embeddings using siamese bert-networks. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, 2019. **Proceedings...** Association for Computational Linguistics, 2019.

REINKE, U. State of the art in translation memory technology. **Language Technologies for a Multilingual Europe; Rehm, G., Stein, D., Sasaki, F., Witt, A., Eds**, [S.l.], p.55–84, 2018.

ROBERTSON, S.; ZARAGOZA, H. et al. The probabilistic relevance framework: bm25 and beyond. **Foundations and Trends® in Information Retrieval**, [S.l.], v.3, n.4, p.333–389, 2009.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval**. [S.l.]: Cambridge University Press Cambridge, 2008. v.39.

SELLAM, T.; DAS, D.; PARIKH, A. P. BLEURT: learning robust metrics for text generation. **arXiv preprint arXiv:2004.04696**, [S.l.], 2020.

SHIWEN, Y.; XIAOJING, B. Rule-based machine translation. In: **Routledge encyclopedia of translation technology**. [S.l.]: Routledge, 2014. p.186–200.

SOMERS, H. Machine translation and minority languages. In: TRANSLATING AND THE COMPUTER 19. **Proceedings...** [S.l.: s.n.], 1997.

STEINBERGER, R. et al. DGT-TM: a freely available translation memory in 22 languages. In: INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC'2012), 8., Istanbul. **Proceedings...** [S.l.: s.n.], 2012. p.21–27.

TANG, Y. et al. Multilingual translation with extensible multilingual pretraining and finetuning. **arXiv preprint arXiv:2008.00401**, [S.l.], 2020.

TIEDEMANN, J. Parallel Data, Tools and Interfaces in OPUS. In: EIGHTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC'12), Istanbul, Turkey. **Proceedings...** European Language Resources Association (ELRA), 2012. p.2214–2218.

TOUVRON, H. et al. Llama: open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, [S.l.], 2023.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, [S.l.], v.30, 2017.

Virtaal. **Virtaal - Virtaal, a feature rich translation tool that allows you to focus on translation, without the tool getting in the way.** 2024.

WANG, L. et al. **Multilingual E5 Text Embeddings**: a technical report. 2024.

WATANABE, S. Tree-structured Parzen estimator: understanding its algorithm components and their roles for better empirical performance. **arXiv preprint arXiv:2304.11127**, [S.l.], 2023.

WOOLSON, R. F. Wilcoxon signed-rank test. **Wiley encyclopedia of clinical trials**, [S.l.], p.1–3, 2007.

ZHOU, Z.-H. **Machine learning**. [S.l.]: Springer Nature, 2021.

ZIEMSKI, M.; JUNCZYS-DOWMUNT, M.; POULIQUEN, B. The United Nations Parallel Corpus. In: INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC'16), 10., Portorož, Slovenia. **Proceedings...** [S.l.: s.n.], 2016.

Appendix

A

Neural Models Comparison

Prior to our study, we carried out an extensive comparison of various multilingual model checkpoints, focusing primarily on the open-source models provided by SBERT REIMERS; GUREVYCH (2019). The main objective was to identify and evaluate the most suitable models for Translation Memory Retrieval, which could then be utilized in our research.

In a typical neural search, there are numerous hyperparameters to explore for all models. These include various similarity metrics such as Inner Product and L2 distance, the effect of using normalization, and the considerations specific to multilingual models, like whether to search and match inputs with the source or target segments. Additionally, there are different model checkpoints to consider.

Despite exploring these numerous variations, the observed performance showed neglectable differences. Table A.1 presents the best results for each model checkpoint. Initially the LaBSE model, introduced by FENG et al. (2020), demonstrated the highest performance in this initial evaluation. This outcome is anticipated, as LaBSE is specifically designed for bitext mining. Consequently, we relied heavily on LaBSE as a first-stage retriever for both the RRF and cross encoders.

Subsequently, the ME5, a list of models tailored for bi-text mining, was released and demonstrated a better result over the LaBSE.

Model	chrF	BLEURT
ME5 Large	0.330	**0.344**
ME5 Large Instruct	0.348	0.340
LaBSE	**0.349**	0.337
multilingual qa mpnet base dot v1	0.316	0.329
multilingual qa mpnet base cos v1	0.313	0.324
paraphrase multi mpnet v2	0.315	0.319
multilingual qa MiniLM L6 cos v1	0.306	0.314
paraphrase multi MiniLM	0.313	0.314
multilingual qa MiniLM L6 dot v1	0.29	0.312

Tabela A.1: Baseline Models Performance on the DGT-TM Dataset: Following the methodology outlined in Section 6.2, we conducted extensive variations in aspects such as the choice of similarity function (L2 vs. Inner Product) and whether to index the source or target sentence, among others. However, due to the marginal nature of these changes, we only show the best results achieved for each model.

B

Cross Encoder Training Details

For each Translation Memory, we trained an XLM-R model using both base and large checkpoints. The training process ran for two epochs with a initial learning rate value is set to $1e^{-5}$. For all cross encoder models training we have used a linear decay scheduler with a warmup step with 1000 examples. We used the Adam optimizer (KINGMA; BA, 2014).

Below, we present the training progress of the KDE4 Cross Encoder. The graph displays both the train and validation losses, alongside the validation accuracy (indicating whether the model correctly ranked the anchor and negative examples). Typically, the train loss decreases rapidly and remains steady throughout the run. However, the validation loss and rank accuracy continue to increase.

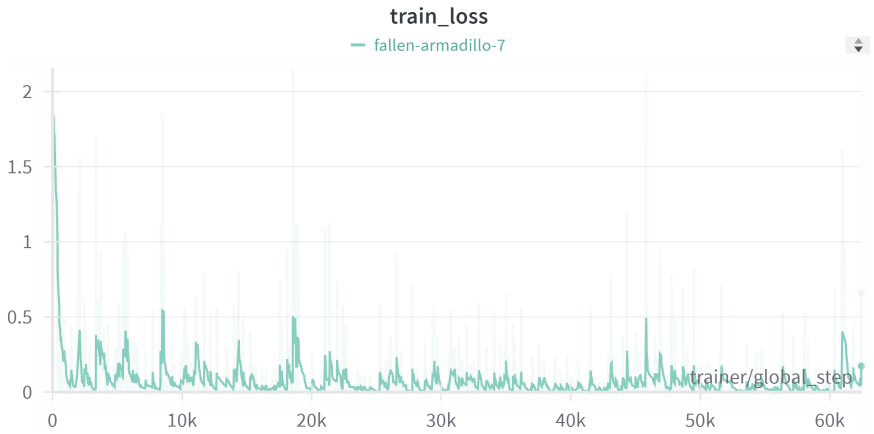

Figura B.1: Train Loss for the KDE4 model.

The Figure B.4 illustrates an example of a problematic training run, where the train loss rapidly escalates. We encountered numerous instances like this earlier on, and found that decreasing the learning rate and incorporating warmup helped mitigate this issue. However, it can still occur in certain scenarios. We hypothesize that this phenomenon occurs due to a combination of inputs that may cause the model to overfit or induce overflow errors during backpropagation. Further investigation is necessary to confirm and better understand the root cause.

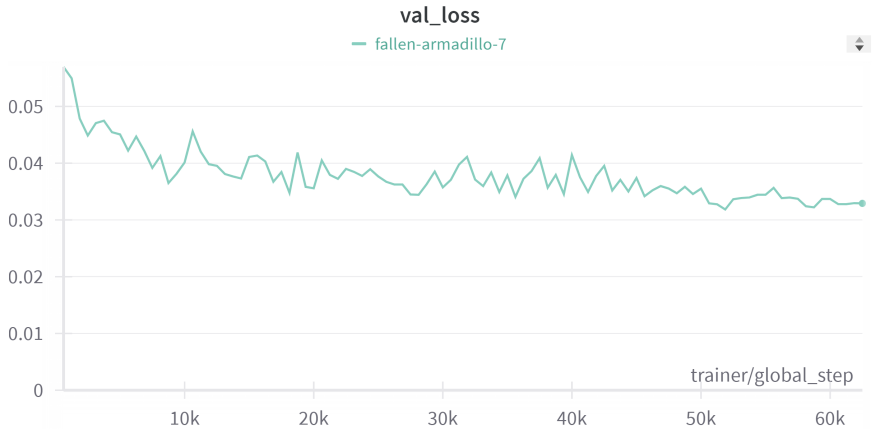

Figura B.2: Validation Loss for the KDE4 model.

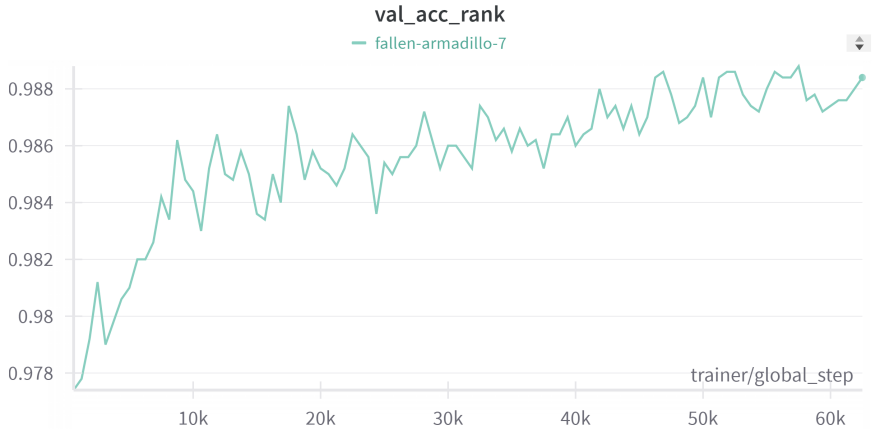

Figura B.3: Accuracy for the KDE4 model.

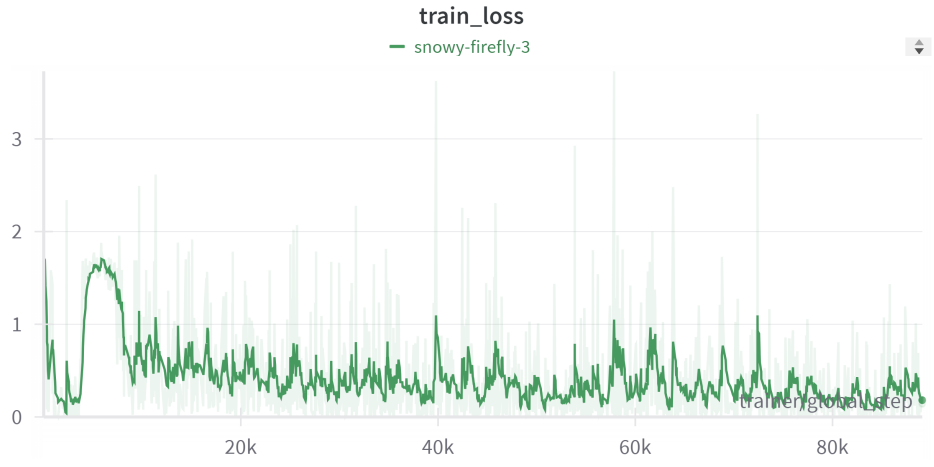

Figura B.4: The graph illustrates an instance where the model followed a wrong optimization path, leading to a significant increase in error, as evident in the curve.

C

Failed Attempts

This chapter introduces some earlier failed attempts in our research. First, we discuss how finetuning models for dense retrieval has not improved performance over baselines. Then we detail how optimizing individual languages indices weights did not improve results in the test set.

C.1 Dense Retrieval Finetuning

TU Retrieval 6.2 is the main bulding block of our RTMR pipeline. So one of the first ideas was to finetune a BERT like model to create specialized representations that could be used in this phase.

The dataset preparation mirrors the approach outlined in the Cross Encoder Section 6.3. Each example is processed through an XML-R model, with the $[CLS]$ embedding serving as the sentence representation

For the training phase, we opted for the Triplet Loss, a popular choice for training neural models to learn representations. Given the $(anchor, positive, negative)$ triplet, this loss works by minimizing the distance between the anchor and positive, while maximizing the distances between the anchor and negative.

Triplet loss is computed as follows, where d is the euclidean distance and α is the c margin:

$$\text{triplet loss} = \max\{d(anchor, positive) - d(anchor, negative) + \alpha, 0\} \tag{C.1}$$

Despite successfully training a model, its performance remained inferior to the USE or LaBSE baselines. We attribute this to the fact that these models were trained on significantly larger datasets, consequently becoming more proficient at representing sentences in the embedding space.

C.2 Optimize RRF Weights

In Chapter 8, we present our findings that demonstrate the state-of-the-art performance of our RTMT approach, surpassing previous baseline methods in the literature. However, there was room for improvement in terms of the language indices selection used for subsequent Rank Fusion. Specifically the Global Voice TM showed that the use of many indices lead to better results, when for the others the source and target obtained the best results.

For this we conducted an experiment to test whether we could use all indices languages, but applying weights for each one. The initial proposed setup was the following.

- Divide the Translation Memory index translations unit into train and dev sets.
- Build the individual language indices (TU Retrieval index phase)
- Now we create a function that receives a new input segment, language indices, a list of weights for each individual rank and it returns the merged rank following our RTMR approach, but applying the weights in the RRF.
- Then we use Optune to optimize the weights for the previous function. The algorithm used for optimization was the TPE (BERGSTRA et al., 2011), (WATANABE, 2023).

Table C.1 provides a comparative analysis of the utilization of weights derived from the optimization step for the weighting of each respective rank. As we can see the optimization did not improve, as the results remain largely unchanged.

Translation Memory	Optimize RRF	RRF Many Indices
DGT-TM	0.332	**0.332**
United Nations	0.331	**0.331**

Tabela C.1: The results from the Optimize RRF againts using RRF with no weights. Both methods used all the avaliable languages in the Translation Memory