

**ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA**

**DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING**

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Natural Language Processing

**IMPLEMENTING LARGE LANGUAGE
MODEL-BASED MACHINE TRANSLATION
IN SMALL AND MEDIUM-SIZED
ENTERPRISES**

CANDIDATE

Álvaro Esteban Muñoz

SUPERVISOR

Prof. Paolo Torroni

CO-SUPERVISOR

Dott. Gianmarco Pappacoda

Academic year 2023-2024

Session 1st

A mi querida familia, que, aunque no siempre estuvo de acuerdo con mis decisiones, siempre supo apoyarlas.

Gracias de corazón por confiar en mí.

Contents

1	Introduction	1
1.1	Definition of Machine Translation	1
1.2	Motivation	2
1.3	Aims and Objectives	3
1.4	Document structure	4
2	Theoretical Framework	6
2.1	Bayes' Theorem	6
2.2	The Machine Translation Task	6
2.3	Machine Learning Paradigms	9
2.4	Large Language Models	9
2.5	Language Clustering	11
2.6	Low-Resource Languages	12
2.7	Metrics for Machine Translation	13
2.7.1	BLEU	13
2.7.2	ROUGE	13
2.7.3	Embedding Similarity	14
2.8	Technologies used	14
3	Approach	17
3.1	Problem Description	17
3.2	Machine Translation Engine	18
3.2.1	Model considerations	19

3.2.2	Memory Considerations	22
3.3	Dataset Creation	23
3.3.1	From TMX to CSV	23
3.3.2	Web Scraping from EUR-lex	24
3.3.3	From HTML to CSV	25
3.4	Metrics	26
3.4.1	Automatic Evaluation	27
3.4.2	Manual Evaluation	27
3.5	Machine Translation API	27
3.6	Machine Translation User Interface	28
3.7	CAT Application Plugin	28
4	Results	31
4.1	Automatic evaluation	31
4.2	Manual evaluation	33
5	Conclusion	35
5.1	Limitations	35
5.2	Future work	36
	Bibliography	37
	Acknowledgements	48

List of Figures

3.1	MT System Architecture	18
3.2	MT Engine pipeline	20
3.3	Language families inside indoeuropean languages	22
3.4	Cross-lingual text alignment	26
3.5	Sample sheet given to experts for manual evaluation	28
3.6	MT User Interface	29
3.7	Configuration panel for the custom MT engine	30

List of Tables

3.1	Language cluster codes	22
3.2	Dataset size for each language pair/cluster and pre-trained models associated to each of them	24
4.1	Results of the models on the test set	33
4.2	Average grading given by the experts for each language pair	34

Abstract

With the release of Large Language Models (LLMs), namely the GPT models, many companies have integrated AI-based technologies to automate natural language tasks like summarization, question-answering, and translation. However, Small and Medium-sized Enterprises (SMEs) face a significant challenge in leveraging these advancements due to limited resources. Unlike large corporations (e.g., Google, Meta, or Amazon), SMEs often lack not only the computational power and financial capacity to train LLMs from scratch but also the vast amounts of data that they require for training, forcing them to rely on external models or services.

This work addresses the problem of implementing a machine translation (MT) system tailored for an SME, Medhiartis s.r.l., with limited resources. Our approach involved fine-tuning pre-existing LLMs using the company's proprietary data to create customized translation models. We systematically evaluated these models' performance and developed an API to integrate them into a functional MT pipeline. The API was deployed in two applications: a plugin for a Computer-Assisted Translation (CAT) tool and a web-based translation interface, both designed to streamline translation tasks for the company.

This study demonstrates how SMEs can effectively adapt LLMs to their specific needs, providing a practical solution for high-quality machine translation in resource-constrained settings.

Chapter 1

Introduction

1.1 Definition of Machine Translation

Machine translation is area of computational linguistics which involves the automatic production of a target-language text on the basis of a source-language text [19]. The critical role of computers during the Second World War led to the rise of various non-numerical applications. Among these early applications, one of the pioneering developments was machine translation. Automatic translation was mainly operating in defence under government and international organisations by the 1960s and 1970s. It won't be until the end of the century when these applications will start to enter in commercial settings [19].

Machine Translation is usually handled using rule-based, statistical or neural approaches [36]. However, in the last few years, neural approaches using LLMs have grown drastically [36], in fact, on this project we are going to focus on this last approach and how a Low-resource (LR) environment, such as SMEs, can manage to access this so powerful language models, mainly relying on something called LLMs Operations (LLMOps). LLMOps is the name given to the set of best practices, techniques and tools used for the operational management of large language models in production environments¹.

¹MLOps-databricks

1.2 Motivation

Machine Translation (MT) is a well-established field that has undergone many periods of evolution as well as phases of inactivity. The earliest developments were started by Rule-Based Machine Translation (RBMT), which consisted of human-crafted linguistic rules based on grammar, syntax, and vocabulary knowledge [14, 43]. This approach was later succeeded by Statistical Machine Translation (SMT), which relied on big datasets of bilingual text to generate translations based on probabilistic models [22, 6]. Despite its limitations, SMT was the dominant approach for many years [43]. In recent years, the rise of Deep Learning has brought Neural Machine Translation (NMT) to the front as the prevailing approach for implementing translation systems. Unlike RBMT and SMT, NMT employs artificial neural networks to model translation patterns.

Early NMT systems were based on Recurrent Neural Networks (RNNs), which processed sentences sequentially and captured contextual information over time [55]. However, the development of the Transformer architecture by Vaswani [51] has become the standard way of addressing this problem. The transformer architecture introduced attention mechanisms that allowed models to focus on different parts of the input sentence dynamically, addressing many of the weaknesses of RNN-based systems, such as their inability to model long-range dependencies effectively. This rapidly became the standard for NMT and enabled the rapid advancement of machine translation technologies, setting the foundation for state-of-the-art systems such as OpenAI's GPT models and Google's continued advancements in the field [51, 4].

Current directions in Machine Translation (MT) are spreading into several areas, including Low-resource NMT, which focuses on developing translation systems for environments with limited linguistic resources [16]. Another emerging trend is dealing with informal spelling, such as colloquial expressions, slang, and typographical inconsistencies, which are common in social

media and casual communication [49, 2]. Additionally, there is growing emphasis on deploying NMT in more and more applications, such as document-level and speech translation [54, 11].

There are many NMT technologies available that do not require building an entire translation system from scratch, like Google Cloud Translation² or Azure AI Translator³. However, these technologies come with two significant disadvantages: I) they create dependency on the company that owns the technology, and II) they often compromise data privacy. Relying on another company's translation service means your data must be shared, which is typically unwanted by companies. This project shows a more accessible approach suitable for Low-resource environments, such as SMEs, based on the guidelines and best practices provided by many companies like Databricks⁴ or Weight and Biases⁵.

1.3 Aims and Objectives

This project aims to explore the feasibility and effectiveness of implementing a LLM-based machine translation system within a small company, therefore facing the issue of limited resources. The following goals have been considered in pursuit of enhancing translation efficiency and effectiveness, minimizing costs, mitigating risks, and ensuring feasibility and scalability.

Exploration of benefits

- Evaluate how our solution helps the company's needs related to the translation.
- Assess possible improvements in efficiency, cost and effectiveness compared to company's traditional methods.

²<https://cloud.google.com/translate>

³<https://azure.microsoft.com/en-us/products/ai-services/ai-translator>

⁴<https://www.databricks.com/>

⁵<https://wandb.ai/>

Analysis of feasibility

- Explore the computational demands of our approach and how the company can implement it with its current resources.

Evaluation of Translations

- Measure a set of metrics which tell us how good is performing our solution.
- Assess potential downsides and limitations of our approach.

Develop and implementation

- Create a dataset suitable for model's training from company's data.
- Train and evaluate a set of models for automatic translation.
- Fit the models inside a system accessible to the company.

1.4 Document structure

This document is structured into five chapters listed below:

Chapter 1: Introduction The first chapter provides a general overview of the motivation behind the project, focusing on the challenges SMEs face in adopting LLM-based approaches, namely for machine translation. It outlines the research objectives and scope of the project.

Chapter 2: Theoretical Framework This chapter provides the theoretical foundations of machine translation and Large Language Models (LLMs). It reviews relevant literature on neural machine translation, the development of LLMs, and their applications, providing context for the techniques used in this work.

Chapter 3: Approach This chapter outlines the methods and techniques applied to develop the machine translation system for Medhiartis s.r.l. It describes the dataset preparation, model fine-tuning, system architecture, and describes the evaluation method along with the used metrics. In addition it also outlines different considerations taken into account for the chosen models.

Chapter 4: Results In this chapter, the performance of the trained models is evaluated through various metrics introduced in the previous chapter. The effectiveness of the system is assessed, and comparisons with results given by experts on the field are made to highlight the system's strengths and areas for improvement.

Chapter 5: Conclusion The final chapter summarizes the key findings, discusses the limitations of the study, and suggests potential future improvements.

Chapter 2

Theoretical Framework

2.1 Bayes' Theorem

Before discussing probabilistic language models, it's important to provide a brief introduction to probability theory, particularly Bayes' Theorem. Bayes' Theorem is a mathematical rule that allows us to determine the probability of a cause A given its effect B . This is known as the conditional probability of A given B , denoted as $P(A|B)$. The theorem is expressed mathematically as:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (2.1)$$

By inverting the conditional probabilities, we can calculate the likelihood of A given B . While Bayes' Theorem is commonly used in statistics, it also serves as the foundation for probabilistic language models. These models estimate the probability of a word occurring in a sentence based on the preceding words [18].

2.2 The Machine Translation Task

Machine translation is a crucial task within the field of NLP, especially for reading newspaper articles or gathering information from online sources like

Wikipedia or government websites in different languages. This makes it one of the most widespread applications of NLP technology [18]. Human language is incredibly complex, and globalization has heightened the need for accurate translation. Today, understanding other languages is more important than ever [21].

Rule-based The history of MT starts with rule-based (RB) systems which was the dominant approach until a few years ago [36]. RB-MT deals with translation by means of static rules which determine how to act on each situation. The main problem of this technique was its exponential growing in system's complexity due to the difficulty of developing a set of rules which covers all the possible vocabulary and syntactic structures that can be done inside a language. The cost of developing a good rule-based machine translation system was quite high and it required highly skilled linguistics on the field to do so [19] [48].

There have been multiple approaches to RBMT through history. Inside these approaches we can find **Direct Translation Systems**, which translates directly by means of rules. An example of this approach is the so called "Dictionary-based" translation, which translates word by word like a dictionary [33].

Transfer-based approaches are also an important group inside RBMT. These approaches analyse the source language to find its grammatical structure and transform it to a new one which suits better the target language for its subsequent translation. [15].

A more representation-dependant approach is the **Interlingua Translation**, which directly translates into an abstract language or representation, facilitating the consequent translation into the target language [1].

Statistical Entering deeper in the current century we can observe the increasing interest in data-driven approaches, namely here we enter the realm

of statistical machine translation [19]. Statistical MT aims to build a probabilistic language model from the observed data. One fundamental concept in SMT is the use of n-grams, which are sequences of 'n' consecutive words from a text. By analyzing these n-grams, the model computes the probability of word sequences across the entire vocabulary [29]. For instance, if we have the bigram (2-gram) "I like" it is most likely for the next word to be "pizza" than "over". For the same reason, we expect $P(T|S)$ to be lower for a pair like $P(\text{Mi piace la pizza}|\text{I am a musician})$ and higher for a pair like $P(\text{Mi piace la pizza}|\text{I like pizza})$, where $P(T|S)$ is the probability of a translator generating T given S [6]. Common approaches to SMT are Phrase-based, by translated whole blocks learnt from parallel corpora [28], or Syntax-based, using parse trees to translate syntactic units instead of words or phrases [57]. The two last mentioned approaches gave birth to a hybrid one called *Hierarchical phrase-based translation* by means of synchronous context-free grammars which is learnt from parallel text with no syntactic notations [7]. There are many issues related to SMT, one of them is the word alignment problem, which consists on how to define the correspondence between the words of the source sentence and the words of the target sentence. A famous approach was covered by Vogel et al. with its HMM-Based Word Alignment model [52].

Neural In 2016, Stanford University developed a Neural MT (NMT) system that significantly outperformed all statistical MT (SMT) systems [26, 27]. Compared to n-gram models, neural language models can process longer word sequences, generalize more effectively across similar contexts, and achieve greater accuracy in word prediction [18]. In practice, a NMT and an SMT perform the same duty, they create a probabilistic language model from the observed data, the difference relies on the complexity of the internal representation created by the system. To build the language model, NMT uses a neural network which consists of many perceptrons [39], analogous to biological neurons, and the activation of many of these units represents a word,

enabling the model to understand complex language structures [19]. At the beginning, neural language models (NLM) used fixed-length of a feature vector to represent each word [58], this representation has been enhanced over time to what we know today as *embeddings*, which are dense, continuous vector representations of words or phrases that capture semantic relationships by positioning similar words close to each other in a vector space.

2.3 Machine Learning Paradigms

To understand how language models are trained, it is important to introduce the most common machine learning paradigms used for this purpose.

Supervised Learning (SL) In supervised learning, both input and output data (labels) is provided to the model during training. The labeled data acts as the "supervisor," guiding the model's learning process [42].

Unsupervised Learning (UL) In contrast, unsupervised learning does not require labeled data. The model learns patterns or relationships within the data without explicit guidance from labels. This approach is commonly used for tasks like clustering and association [42].

Semi-supervised Learning (SSL) Semi-supervised learning combines aspects of both supervised and unsupervised learning. In this case, the dataset contains both labeled and unlabeled data. A typical SSL technique involves training the model on the labeled data, using it to make predictions on the unlabeled data, and then retraining on the newly labeled dataset [42].

2.4 Large Language Models

The main problem of supervised data-drive approaches is the necessity of datasets annotated by humans. It was in the last few years where we discover

that it was possible to train language models on natural language understanding (NLU) by means of self-supervised learning and then fine-tuning those models in more downstream tasks [17, 35, 8]. All of this, along with the introduction of the **Transformer** architecture by Vaswani [51] allowed for much bigger language models, what we call today *Large Language Models* [59]. When we say *Large Language Models* we refer to Language Models, not necessarily based on transformers, possessing ten to hundreds of billions of parameters that are trained on large amounts of data [30]. The main tasks used for general NLU are namely Masked Language Modeling (MLM) [9] and Causal Language Modeling (CLM) [37], which are **self-supervised learning** techniques. In contrast to the high costs associated with manual annotations in strong supervised learning, the self-supervised paradigm avoids these expenses. This approach enables the automatic generation of annotations directly from the data itself [3].

Masked Language Modeling (MLM) Imagine you have the following sentence:

”The dog is playing with the frisbee”

then you could ”mask” one of the words and train your model on learning the correct word that should replace the mask:

”The dog is playing <MASK> the frisbee”

clearly, masking sentences is a process that can be easily done in an automatic way, therefore we could scale this process by taking vast amounts of text data contained in the internet and masking the different existent sentences. The main example using MLM for language modeling is Google’s BERT model [9].

Causal Language Modeling (CLM) As for MLM, CLM is a task that can be easily automated. The concept is similar, the final part of a sentence is

truncated, and the model learns the probability distribution based solely on the tokens preceding the predicted one. This creates a unidirectional context window. CLM-based models are particularly well-suited for text generation tasks. Moreover, they excel as few-shot learners, demonstrating the ability to make highly accurate predictions with only a small number of labeled samples [30].

”I am going to study → to the library”

in this case, ”to the library” is the part of the sentence which needs to be generated by the model. GPT family of models is an example of models trained using CLM [37].

The key difference between the two tasks lies in the context the model uses to learn the probability distribution. A model trained using MLM utilizes all tokens surrounding the masked token, whereas a model trained using CLM focuses only on the tokens in the left context, predicting tokens on the right. When this approach is applied sequentially, it is referred to as autoregressive language modeling. The first approach (MLM) is capable of constructing more robust and accurate language representations. However, the repetitive mask-and-predict process makes the model significantly less efficient when applied to unsupervised tasks [41].

2.5 Language Clustering

Machine translation has to deal with one important problem, it has to handle a vocabulary in two or more languages. For small systems is fine to have one model for each language pair, however, this becomes a problem when scaling up. Imagine you want to develop a model for all the 24 different official languages on the E.U. We would need to train $C^R(24, 2) = 300$ different models, where C^R represents the number of permutations (combinations

with order). This is clearly unfeasible. Even if we consider using a model trained on one language pair for bidirectional translation, we would still need to train $C(24, 2) = 276$ models, where C represents the number of combinations (without regard to order).

The discovery of Large Language Models lead us to think of a possible model able to learn all languages in the world, the so called multilingual language models [60]. Nevertheless, according to the World Atlas of Languages¹ (WAL) from UNESCO, there are around 7000 currently spoken different languages in the world and even discarding the minority languages we would still be left with roughly 500, which is still unfeasible. A hybrid approach to this problem was proposed by Tan et al., **Language Clustering** [44]. The idea is actually quite simple, languages from the same family might have a similar internal representation into a neural network, therefore we would be able to cluster languages from the same family in one model without losing too much accuracy when translating from one into another [44]. This type of language clustering is done using "prior knowledge", however, we could also apply clustering algorithms over the language embeddings.

2.6 Low-Resource Languages

Machine Translation typically requires large amounts of parallel data for the languages being translated, but this is not always possible for many of the over 7,000 languages spoken worldwide. A significant amount of textual data remains unlabeled, noisy, or entirely unavailable for certain languages. This brings us to the concept of Low-Resource Languages (LRLs), which refers to languages that lack sufficient resources. A language can be considered LRL either due to the scarcity of resources for the language itself or for specific domains.

¹WAL-UNESCO

Defining an LRL is challenging [16], as the criteria, such as the number of native speakers or the availability of datasets, can vary. Moreover, this definition evolves over time, a language considered low-resource today may not be in the future [38, 16]. LRLs have become a focus of research in recent years due to the economic and industrial benefits of automatic translation, particularly in countries with multiple official languages like India and Spain [38].

2.7 Metrics for Machine Translation

The evaluation of Machine Translation task is not trivial, we cannot just rely on simple automatic metrics, a manual review is necessary in most of the cases. Even though, these metrics can give us a good starting point or a way of monitoring which models are improving with respect to a baseline.

2.7.1 BLEU

The Bilingual Evaluation Understudy (BLEU) [34] compares a translation to a gold standard in terms of n-gram precision. Although it relies only on syntactical features, i.e., n-grams occurrence, to evaluate the translation, it is very powerful due to its ability to use more than one reference as gold standard, this means, the more translations we have as a reference, the more accurate will be our metric and the more robust will be to semantic discrepancies like synonyms or expressions.

2.7.2 ROUGE

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [24] is a set of metrics widely used for evaluating machine translation. These metrics assess the overlap of n-grams between the machine-generated text and a reference, with different ROUGE score types focusing on specific aspects of this

overlap. In this project, we rely on ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-LSUM.

ROUGE-N (including ROUGE-1 and ROUGE-2) measures the precision and recall of overlapping n-grams, with "N" indicating the length of the n-grams being compared. ROUGE-L, on the other hand, is not constrained by a specified n-gram length but instead leverages the Longest Common Subsequence (LCS) between the texts. ROUGE-LSUM is a variation of ROUGE-L that divides the text into sentences, computes the LCS for each sentence, and then averages these results to produce a score for the entire piece of text.

2.7.3 Embedding Similarity

Text similarity can be approached in various ways, though not all methods focus on semantic similarity. A common semantic approach involves computing vector representations of the sentences in the same vector space, also known as embeddings, and then calculating their cosine similarity [53]. The method used to compute these embeddings can vary, but in the context of machine translation, it is crucial to employ a model able of embedding texts in different languages into the same space. An example of such a model would be a multilingual version of BERT [10].

2.8 Technologies used

This section outlines the set of technologies used in the project. Given the project's magnitude, a diverse range of components had to be implemented, necessitating the use of various technologies. The main goal was to select the most suitable technology for each component, thus ensuring optimal performance and functionality throughout the project.

Python For both the training and evaluation of machine learning models, we opted to use the Python language, which is very popular for this such type

of tasks due to its simplicity and the abundance of libraries existing for it.

Google Colaboratory Training the different machine translation engines requires substantial computing power, therefore we had to depend on companies supplying it. For this project we decided to use Google Colaboratory which is a hosted Jupyter Notebook service that provides access to computing resources like GPUs and TPUs.

HuggingFace Training big models like LLMs from scratch is a challenging task, thus we rely on transfer learning and fine-tuning techniques to achieve so. Additionally, training and evaluation loops vary a lot depending on design decisions. The Hugging Face API furnishes us with a multitude of pre-trained models which are available for fine-tuning, as well as a lot of useful classes facilitating the training and evaluation of models.

JavaScript For two main purposes, we opted for JavaScript:

- Front-end development
- Web scraping

The ability of the language to work with web elements simplifies the design of the web application's front-end and the web scraping part to extract data to train the models.

React The integration of the React library significantly contributed to achieving a more streamlined and modular design for the web user interface. React's component-based architecture facilitated the development process by allowing for the creation of reusable UI elements. Additionally, React's virtual DOM and efficient rendering mechanism contributed to enhanced performance, this approach not only simplified the design process but also improved the overall responsiveness and scalability of the web application.

Selenium For the web scraping of textual data we employed the Selenium library. This tool enables us to run headless browsers which navigates through the websites autonomously looking for the needed information.

Flask The Flask framework for python was the ideal option for the creation of our API for the machine translation engines. Its simplicity allowed an easy integration with the rest of the components, allowing us to create a lightweight API in just a few lines of code.

Chapter 3

Approach

3.1 Problem Description

This project aims to develop a machine translation system based on Large Language Models (LLMs) without depending on APIs of commercial ones. When we say "machine translation system", we discuss every component that makes a machine translation application to work inside the company, which in this case can be summarized as follows:

- Machine Translation engine
- Machine Translation API
- Web application user interface
- Computer-Aided Translation Application Plugin

The whole systems can be seen on figure 3.1.

In the following sections we will cover the approaches applied for the implementation of each component.

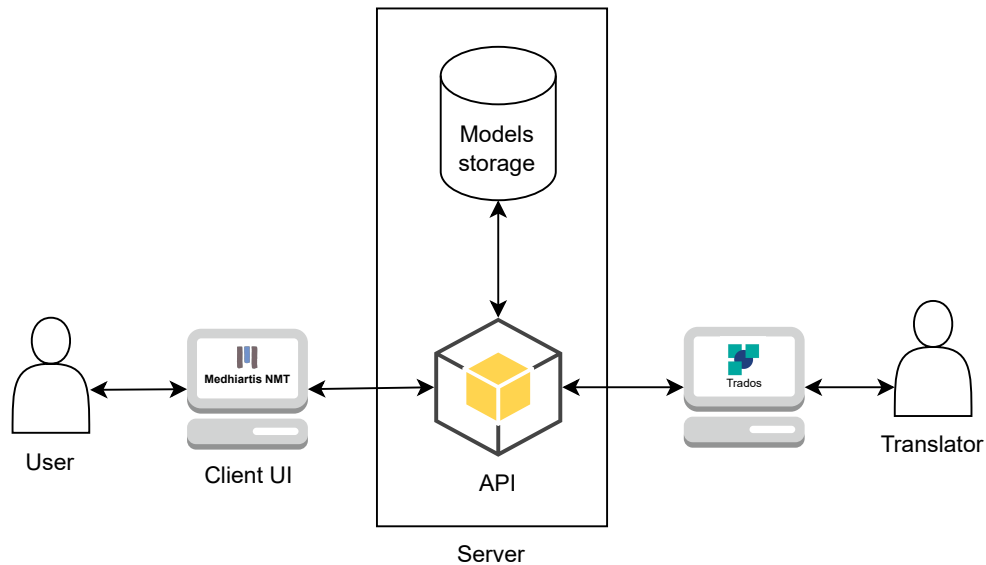


Figure 3.1: MT System Architecture

3.2 Machine Translation Engine

During the development of the engine we followed some of the best practices and guidelines for Large Language Models Operations (LLMOps), provided by Weight and Bias¹. The first important decision to take when implementing a system based on LLMs is whether to train a model from scratch is suitable for our company². There are three basic approaches:

1. Use the API of a commercial LLM.
2. Use an existing open-sourced LLM.
3. Pre-train an LLM from scratch.

The company considers the provided data for training the models to be sensitive, which excludes option 1. This leaves us with only options 2 and 3. However, option 3 demands significant computing resources, substantial financial investment, and considerable time to train large models, making it a challenging approach for small and medium-sized enterprises (SMEs). Given

¹<https://wandb.ai/>

²LLMOps-wandbai

that the company lacks sufficient data and computing infrastructure, this option becomes unfeasible. As a result, focusing on the second option becomes the most practical choice. The open-source community is rapidly expanding, and as large language models (LLMs) gain popularity, they are increasingly being optimized and pre-trained by the community, allowing SMEs to leverage them for commercial purposes. A diagram of the following pipeline for the MT Engine development can be seen in figure [3.2].

3.2.1 Model considerations

The choosing of a good pre-trained model is a core part of this approach and it depends mainly on the amount of data we have and the languages we want to cover within the same model. As a general rule, for multilingual models we will use bigger pre-trained transformers. The first models chosen were the mBART [25] and mT5 [56] where "m" stands for multilingual. These two models are pre-trained on the fill-masking for natural language understanding, hence they have to be trained on the downstream task on the translation task through the data given to them (fine-tuning). Notice that, due to the huge number of parameters these models possess, 680M for mBART and 1.2B for mT5 on its large version [25, 56], the training process becomes significantly slower.

While data availability is not an issue for languages like Italian, Spanish, French, or English, it becomes a significant challenge for others. The company lacks enough data for languages such as Arabic or Chinese, and the situation is even worse for languages like Czech and Slovene. At this point, it is important to address a key issue with Low-Resource Languages (LRLs). Since the models are trained solely on data produced by the company, the limited availability of data for the mentioned languages arises mainly from the fact that they are not in high demand by the company's clients. In this context, these languages can be said to be low-resource within the company's domain. This is likely

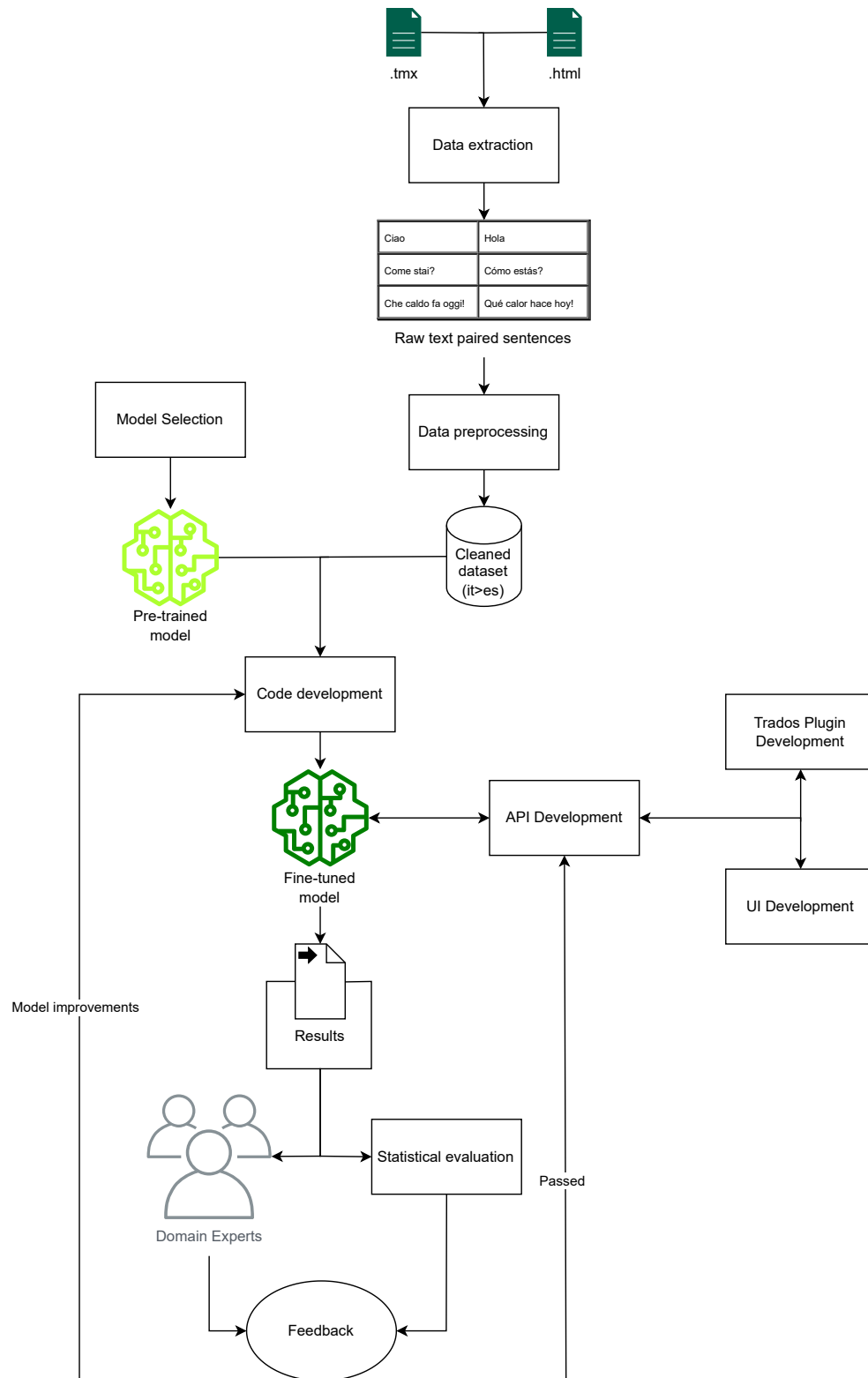


Figure 3.2: MT Engine pipeline

due to the geographical focus of the company's business, which is in Italy and neighboring countries. While this explains why languages like German have more resources available than Chinese, it does not apply for languages like Slovene. Languages such as Slovene and Czech are inherently low-resourced, meaning they have limited resources and are often not considered in the development of language technologies [38]. Machine translation for low-resource languages is a vast field, with many approaches available to address this challenge [16]. In this project, we explored two strategies: first, we leveraged the power of transfer learning by fine-tuning models, namely MarianMT [46, 47] and SMaLL100 [31, 32], that are already pre-trained for our task in various languages, and second, we search for additional data sources from the internet.

Opus-MarianMT

The Opus-MarianMT³ is a project conducted by the university of Helsinki in which they trained the MarianNMT⁴ by the Microsoft translation team on the Opus⁵ (Open Parallel Corpora). They do not only cover many bilingual language pairs but also cover language clusters [2.5], for example, the model opus-mt-itc-itc is trained in translating between any pair of italic languages. While these models cover important language pairs such as Italian-German and Italian-Arabic, they do not support others, like Italian-Slovene. However, by stepping back in the linguistic classification, we can use the model for the relevant language cluster, for instance, the Indo-European cluster for Italian-Slovene. An example of language clustering inside indo-european languages can be seen on figure [3.2] (italic languages are marked in mustard color)

A list with all the initials used by Helsinki per each language cluster is listed on table [3.1], notice individual languages are not listed since they follow the two letter system of ISO 639⁷

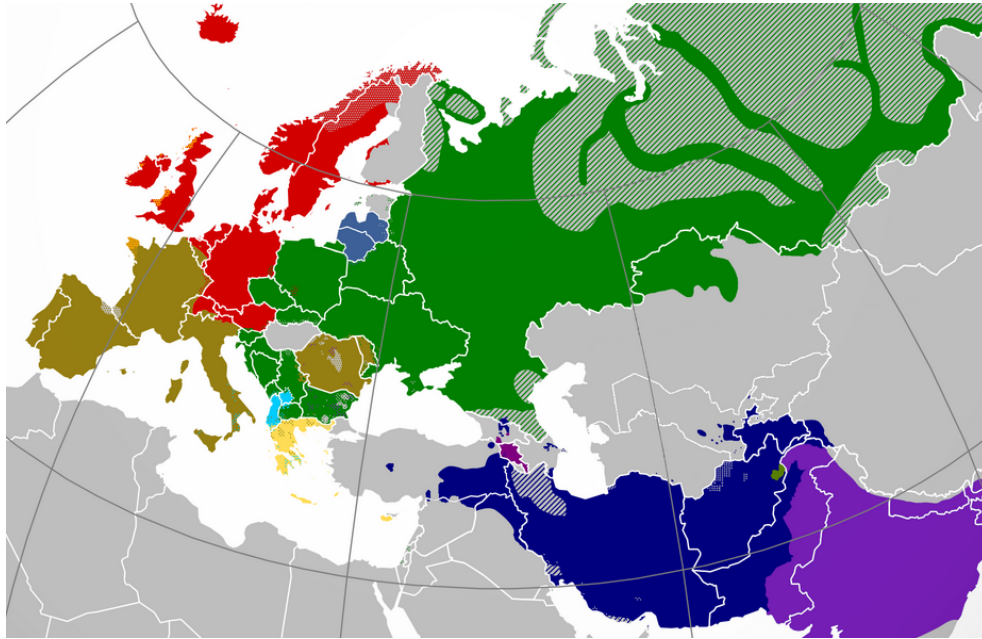
³<https://github.com/Helsinki-NLP/Opus-MT>

⁴<https://marian-nmt.github.io/>

⁵<https://opus.nlpl.eu/>

⁶https://en.wikipedia.org/wiki/File:Indo-European_Language_Family_Branches_in_Eurasia.png

⁷https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes

Figure 3.3: Language families inside indoeuropean languages ⁶

Prefix	Full name
ine	Indo-european
itc	Italic
gmq	North Germanic
zle	East Slavic

Table 3.1: Language cluster codes

SMaLL-100

The SMaLL-100 model is a multilingual MT model based on the architecture of M2M-100 by Facebook AI [12] and trained on the Tatoeba dataset by Helsinki-NLP [45]. We rely on this model for those languages that are not covered by Opus-MarianMT which are Simplified Chinese and Hungarian.

3.2.2 Memory Considerations

Multilingual BART and T5 are models trained on general Natural Language Understanding (NLU). Hence, they are really big models that can be down-streamed for many Natural Language tasks. Due to their size, the most convenient thing to do is to downstream them to translate into bigger clusters of

languages like 3 or 4 instead of just a pair.

While an average mT5 model demands 5GB of memory, a MarianMT model averages 2GB per language pair for three pairs. Thus, employing three MarianMT models would require 6GB of memory, which is comparable to the memory needs of an mT5. It's important to note that using separate systems for each language pair enhances scalability, as updating the weights for only one language pair eliminates the need to adjust all pairs. On the other hand, relying on a single system offers the advantage of compactness, making it easier to manage than handling three separate systems. Additionally, inference time is generally faster for a smaller system.

3.3 Dataset Creation

When handling a company's data, it's essential to recognize that raw data isn't directly compatible with a machine learning model. Instead, we must organize a comprehensive **dataset** that consolidates all the raw data from the company. This data may be unstructured and scattered across different files and formats. In this case, the company stores translations in Translation Memory eXchange (TMX) files, an XML-based format used for managing translation memories⁸. However, TMX files contain a significant amount of information that is irrelevant for training purposes. Therefore, we extracted the essential translation units and exported them as a CSV file containing paired translated segments.

3.3.1 From TMX to CSV

First, it is important to define what a Computer-Aided Translation (CAT) application is. A CAT tool is a software designed to assist in the translation of documents by integrating various functionalities. These typically include terminological management, the storage and reuse of previously translated segments through translation memories, and text alignment for consistency and

⁸<http://xml.coverpages.org/tmxSpec971212.html>

efficiency [13].

All company’s data from translated sentences is contained in **translation memories**, which are TMX files generated by SDL Trados Studio, a software for computer-aided translation (CAT). These TMX files are structured in an XML format which allows us to extract the **translation units** specifying the sentence pair created by the translator. All the extraction from the TMX files was performed using a Python script specifically designed to extract the XML tag containing the translation units with the sentence pairs. Sentence pairs are then exported into CSV files, which will be merged to create our final dataset. Data size and pre-trained models used for each set can be seen on table [3.2]

Dataset Type	Source Lang	Target Lang	N° Samples	Pre-trained Model
Multilingual	Italian, English	Italian, English, French, Spanish	520725	mBart-large
Multilingual	Italian	Danish, Finnish, Swedish	554150	mT5-large
Bilingual	Italian	Norwegian	79455	opus-mt-ine-ine
Bilingual	Danish	Italian	266350	opus-mt-tc-big-gmq-itc
Bilingual	Swedish	Italian	89893	opus-mt-tc-big-gmq-itc
Bilingual	German	Italian	480841	opus-mt-de-it
Bilingual	Italian	Dutch	135462	opus-mt-ine-ine
Bilingual	Italian	German	480841	opus-mt-ine-ine
Bilingual	Italian	Portuguese	96615	opus-mt-itc-itc
Bilingual	Italian	Brazilian	32068	opus-mt-itc-itc
Bilingual	Italian	Russian	156247	opus-mt-tc-big-it-zle
Bilingual	Italian	Arabic	27871	opus-mt-it-ar
Bilingual	Italian	Turkish	86490	opus-mt-tc-big-itc-tr
Bilingual	Italian	Czech	13960	opus-mt-ine-ine
Bilingual	Italian	Polish	87815	opus-mt-ine-ine
Bilingual	Italian	Chinese	35082	SMaLL100
Bilingual	Italian	Romanian	9604	opus-mt-ine-ine
Bilingual	Italian	Slovene	10380	opus-mt-ine-ine
Bilingual	Italian	Hungarian	13676	SMaLL100

Table 3.2: Dataset size for each language pair/cluster and pre-trained models associated to each of them

3.3.2 Web Scraping from EUR-lex

As mentioned earlier, the primary issue with the company’s data lies in the lack of sufficient data for languages beyond the most commonly spoken ones. This gap is particularly evident for languages like Ukrainian or Portuguese for example, necessitating the acquisition of additional data sentences. To address this challenge, the approach employed involved sourcing documents from EUR-Lex, an online platform housing official legislative documents of

the European Union, available in all 24 of its official languages [50]. The web scraping was performed thanks to Selenium’s API for the Javascript language. A small script was developed for the purpose of navigating the platform and automatically extract most of the legislative files contained in all the languages.

3.3.3 From HTML to CSV

Extraction of datasets from legal documents involves another Natural Language Task, **Cross-lingual text alignment**. Each nationality has its own regulations for official documents, hence, given two documents in two different languages, sentences may not be aligned, making its automatic extraction much more difficult. To address this problem, we implemented a similarity-based approach. The method involves analyzing each line of every document by maintaining a cursor in both. This cursor progresses through both documents for each sentence pair that exhibits a **sufficiently high semantic similarity**, surpassing a predetermined threshold. If the sentences **are not** sufficiently similar, we measure the semantic similarity of the current left sentence with the preceding, current, and subsequent sentences in the right document [figure 3.4]. The correct right sentence is then determined as the one with the highest semantic similarity value, provided it meets a second threshold, which is either lower or equal to the previous one.

Semantic similarity, when applied to sentences, is better referred to as *Sentence Similarity*. It is computed by bringing both sentences into the same **vector space** and then calculating their cosine similarity. To do this, sentences are first transformed into numerical high-dimensional vectors (embeddings), using a multilingual transformer model, which has been pre-trained explicitly for this purpose. Cosine similarity measures the cosine of the angle between two vectors, with values ranging from -1 (indicating completely opposite vectors)

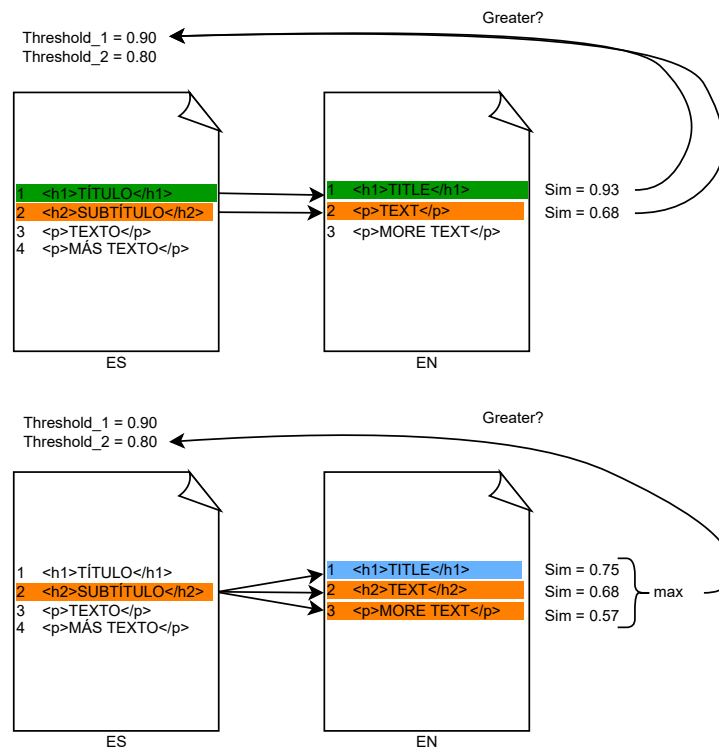


Figure 3.4: Cross-lingual text alignment

to 1 (indicating identical vectors), and 0 meaning the vectors are orthogonal, or unrelated. Semantic similarity using this approach may vary from one language to another, yet, since we are comparing the metric for the same pair of languages all the time this won't be a huge issue. The resulting data was augmented by **7M sentences** for each language, which was reduced to **2.5M sentences** after removing those ones composed by less than 4 words. When compared to the approximately **100k sentences** available for Portuguese, it's evident that the dataset underwent a significant expansion, well surpassing our initial expectations for data enrichment.

3.4 Metrics

After fine-tuning our models, we needed to evaluate their performance. This evaluation was conducted with the assistance of expert translators. However, since manual review is a time-consuming process, we decided to eliminate

the poorest-performing models using automatic metrics before submitting the results to the experts.

3.4.1 Automatic Evaluation

We used automatic metrics such as BLEU, ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-LSUM, and text similarity based on embeddings. It is important to note that we only have one reference sentence for each generated translation, which limits the full potential of the BLEU score. Nevertheless, BLEU, along with the other metrics, provides us with a valuable indication of the model's performance.

For the embedding-based Text Similarity, we used the distilbert-base-multilingual-cased[40] model. This model covers all the languages we are evaluating and can embed all the sentences into the same vector space, allowing us to compute an appropriate cosine similarity.

3.4.2 Manual Evaluation

Manual evaluation is conducted by reliable freelance translators, who regularly collaborate with the company and have a deep understanding of the company's translation needs. They are provided with a set of 50 samples drawn from the test set results. Each sample is manually graded on a scale from 1 to 10, assigned a label of either 'Acceptable' or 'Not Acceptable,' and annotated with possible errors identified in the model's output. An example of these evaluation sheets can be seen in Figure 3.5 (Notice they are not real samples from the dataset).

3.5 Machine Translation API

Once our models are prepared for deployment, the next step involves establishing a mechanism for interaction, which is facilitated through the development

Sr No	Input Text	Output Text	Model Generated Text	Grading (0-10)	Acceptable or Not
1	Mi piacerebbe visitare Roma il prossimo anno.	Me gustaría visitar Roma el próximo año.	Me gustaría visitar Roma el próximo año.	10	Yes
2	Stiamo lavorando su un nuovo progetto molto interessante.	Estamos trabajando en un nuevo proyecto muy interesante.	Estamos laborando en un nuevo proyecto muy interesante.	8	Yes
3	Non ho avuto abbastanza tempo per completare il lavoro.	No tuve suficiente tiempo para completar el trabajo.	No tuve suficiente tiempo para completar el trabajo.	10	Yes
4	Questa sera andremo a cena in un bel ristorante.	Esta noche iremos a cenar a un buen restaurante.	Esta noche iremos a cenar a un buen restaurante.	10	Yes
5	Il museo che vogliamo visitare chiude alle sei.	El museo que queremos visitar cierra a las seis.	El museo que queremos visitar cierra a las seis.	10	Yes
6	Anche se piove, abbiamo deciso di andare al parco.	Aunque esté lloviendo, decidimos ir al parque.	Aunque esté lloviendo, hemos decidido ir al parque.	9	Yes
7	Vorrei sapere quanto costa prenotare una stanza doppia.	Me gustaría saber cuánto cuesta reservar una habitación doble.	Me gustaría saber cuánto cuesta reservar una habitación doble.	10	Yes
8	Ho comprato dei biglietti per il concerto della settimana prossima.	Compré entradas para el concierto de la semana que viene.	He comprado entradas para el concierto de la semana que viene.	10	Yes
9	Il film che abbiamo visto ieri sera è stato davvero emozionante.	La película que vimos anoche fue realmente emocionante.	La película que vimos anoche fue realmente emocionante.	10	Yes
10	Domani mattina dobbiamo partire presto per arrivare in tempo.	Mañana por la mañana debemos salir temprano para llegar a tiempo.	Mañana por la mañana debemos salir temprano para llegar a tiempo.	10	Yes

Figure 3.5: Sample sheet given to experts for manual evaluation

of an Application Programming Interface (API). The API is designed to accept three inputs: a string representing the sentence, the source language code, and the target language code. Utilizing these inputs, the API determines the appropriate model to perform the translation and loads it from the model storage. The sentence is then processed by the selected model to infer its translation. Finally, the API returns the translated string as its output.

3.6 Machine Translation User Interface

A web user interface was developed so to allow translators (and every other member of the company) to use the MT engine in a comfortable web application [figure 3.6]. The designed is highly inspired on similar apps like google translator⁹ and DeepL¹⁰. The layout was designed by the IT branch of the company, Medhit, and implemented by me using Javascript and React Native.

3.7 CAT Application Plugin

To leverage the power of our machine translation system, the company requested the development of an MT plugin for a CAT application. While not a core component of the system itself, this plugin demonstrates its performance in a real-world scenario.

⁹<https://translate.google.es/>

¹⁰<https://www.deepl.com/>

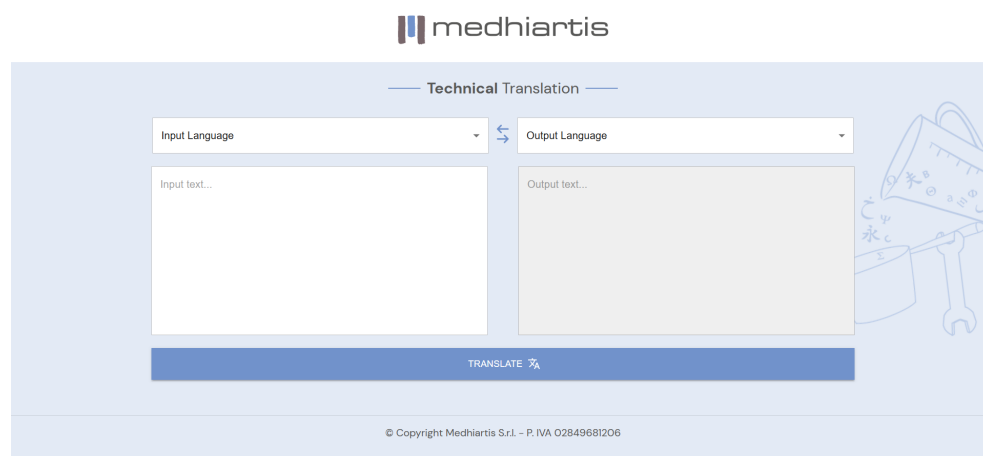


Figure 3.6: MT User Interface

The objective of an MT plugin is to boost the efficiency of translators, allowing them to perform the same job in significantly less time. The plugin achieves this by automatically selecting the appropriate MT model for the given language pair and suggesting a possible translation for the current segment. The translator then has the option to accept or modify the system's suggestion, retaining full control over the final output.

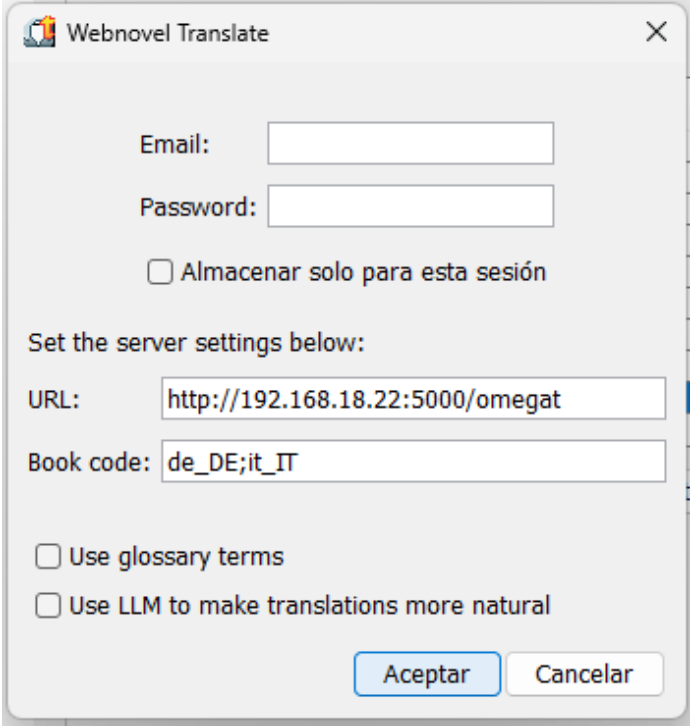
Due to technological constraints, the plugin was originally developed for a different CAT tool than the one used by the company. Specifically, it was created for OmegaT¹¹, a free software for computer-aided translation. The development was based on the OmegaT plugin for custom MT engines¹² initially designed by Atlas Studios¹³. Since the plugin's code is available under the GPL license, it was adapted to meet the specific needs of our company.

The plugin works as follows: it is configured via a panel located in the machine translation settings tab (see figure 3.7). Configuration involves entering the credentials of the company user, the server URL where the API is hosted, and the relevant engine prefixes or *Book code*. For instance, if translating from German (Germany) to Italian (Italy), the configuration would use the format `de_DE;it_IT`, as illustrated in figure 3.7.

¹¹<https://omegat.org/>

¹²<https://github.com/atlas-studios/omegat-plugin>

¹³<http://atlas-studios.com/>



The image shows a configuration dialog box titled "Webnovel Translate". It contains the following fields and options:

- Email: [Empty text box]
- Password: [Empty text box]
- Almacenar solo para esta sesión
- Set the server settings below:
 - URL: `http://192.168.18.22:5000/omegat`
 - Book code: `de_DE;it_IT`
- Use glossary terms
- Use LLM to make translations more natural

At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

Figure 3.7: Configuration panel for the custom MT engine

Chapter 4

Results

4.1 Automatic evaluation

After properly fine-tuning the models, we evaluated the metrics defined in the previous chapter on the test set. The fine-tuning was performed on the dataset with no enrichment (Table 3.2). Results are displayed in Table 4.1. Note that $it > [da,fi,sv]$ and $[it,en] > [en,es,fr]$ represent multilingual datasets, so the results are grouped by model performance on each dataset, rather than by individual language pairs. This means that, although the pre-trained model remains the same in some rows, each row still represents a different fine-tuned model. Key points to highlight from the results are:

mBart-large the main model, mBart-large, reaches a text similarity of 0.87, a really nice result. However, for proper translations we need to avoid being under the 0.90.

Nordic model The results from the model trained for translating Italian into Nordic languages, namely Italian to Danish, Finnish, and Swedish, indicate some challenges. These issues can likely be attributed to two main factors:

1. The model's size is too large compared to the amount of data available for fine-tuning it, which may interfere with its ability to effectively learn

translation patterns.

2. Finnish belongs to the Uralic language family, which is distinct from the Indo-European family to which Italian, Swedish, and Danish belong. Translations among languages within the same family tend to be more straightforward. In contrast, translating to Finnish, which requires learning entirely different linguistic patterns, demands more data. This variance likely contributes to the poorer results for Finnish, negatively impacting the overall performance.

Effectiveness of MarianMT The Text Similarity column reveals that most MarianMT models perform exceptionally well, with the majority achieving scores above 0.93. This indicates that these models are capable of generating translations that closely match the meaning of the original text. However, there are some intriguing points we should underscore. For instance, the Chinese language achieves a respectable text similarity score of 0.87, despite poor performance on BLEU and ROUGE metrics. This discrepancy may be attributed to the isolating nature of Chinese [23], where each morpheme functions as an independent word, allowing for sentences with similar meanings to be constructed from entirely different words or characters. A similar phenomenon is observed in languages like Russian and Arabic, likely due to their highly fusional nature [5]. To be more precise, BLEU and ROUGE are heavily influenced by syntactical variations in word formation, which is why Text Similarity provides a more accurate assessment of a model’s performance. Actually, we can notice that the number of samples for each language pair is directly proportional to the Text Similarity score.

Results on SMaLL100 The results on the last bilingual model, while not on par with MarianMT, are nonetheless significant. The model achieves a score of 0.87 on Chinese and 0.65 on Hungarian, respectable results considering the limited data available for fine-tuning. We can underscore that, unlike

Languages	Pre-trained Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSUM	Text Similarity
[it, en] >[en, es, fr]	mBart-large	43.98	0.57	0.43	0.56	0.56	0.87
it >[da, fi, sv]	mT5-large	22.56	0.54	0.37	0.52	0.52	0.44
it >nb	opus-mt-ine-ine	54.08	0.74	0.58	0.72	0.72	0.98
da >it	opus-mt-tc-big-gmq-itc	83.47	0.91	0.86	0.90	0.90	1.00
sv >it	opus-mt-tc-big-gmq-itc	61.82	0.79	0.67	0.78	0.78	1.00
de >it	opus-mt-de-it	45.62	0.68	0.50	0.66	0.66	0.94
it >de	opus-mt-it-de	49.54	0.67	0.48	0.64	0.64	0.93
it >nl	opus-mt-ine-ine	43.66	0.70	0.50	0.66	0.66	1.00
it >pt	opus-mt-itc-itc	58.75	0.79	0.66	0.78	0.78	0.99
it >pb	opus-mt-itc-itc	61.20	0.81	0.69	0.80	0.80	0.94
it >ru	opus-mt-tc-big-it-zle	42.49	0.29	0.16	0.29	0.29	0.74
it >ar	opus-mt-it-ar	28.10	0.25	0.13	0.25	0.25	0.62
it >tr	opus-mt-tc-big-itc-tr	66.11	0.82	0.73	0.80	0.80	0.99
it >cs	opus-mt-ine-ine	18.72	0.49	0.29	0.47	0.47	0.85
it >pl	opus-mt-ine-ine	38.73	0.65	0.47	0.63	0.63	1.00
it >zh	SMaLL100	25.28	0.36	0.14	0.35	0.35	0.87
it >ro	opus-mt-ine-ine	35.17	0.63	0.44	0.62	0.61	1.00
it >sl	opus-mt-ine-ine	29.75	0.53	0.32	0.51	0.51	0.89
it >hu	SMaLL100	28.32	0.62	0.42	0.57	0.57	0.65

Table 4.1: Results of the models on the test set

MarianMT, SMaLL100-based models require more data to reach a similar performance to those based on MarianMT.

4.2 Manual evaluation

The evaluation provided by translation experts closely aligns with the results shown in Table [4.1]. Their feedback was essential in verifying which models performed as expected. For instance, although the mBart-large model achieved a text similarity score of 0.87, experts pointed out its difficulty in handling accents and special characters, which is critical for technical documentation.

The evaluation was not conducted for every language but was limited to those where the translator was closely associated with the company. These languages included Spanish, English, Italian, French, Swedish, Danish, Norwegian, Finnish, and Russian. A subset of 50 samples from each language pair dataset was used for the evaluation. For instance, the subset provided to experts for the Italian-to-Russian (it>ru) language pair contained 50 translations from Italian to Russian. In contrast, for the Nordic languages, Danish, Finnish, and Swedish, 50 samples were randomly selected from the test set and translated into their respective target languages. As a result, the outcomes

Language Pair	Avg. Grading	Pre-trained model
it > es	8.83	mBart-large
it > fr	7.84	mBart-large
it > en	8.41	mBart-large
it > da	4.16	mT5-large
it > fi	0.00	mT5-large
it > sv	2.70	mT5-large
it > nb	9.63	opus-mt-ine-ine
it > ru	9.27	opus-mt-tc-big-it-zle

Table 4.2: Average grading given by the experts for each language pair

in Table 4.2 are organized by language pair and not by dataset as in Table 4.1, with each row representing the averaged feedback from a translator.

The experts’ feedback supports our hypothesis regarding the mT5-large model’s performance with the Finnish language. While the average score for Swedish and Danish hovers around 4, the average rating for Finnish is significantly lower at 0. Moreover, the results from Marian-based models show a similar trend to those produced by the automatic evaluation.

Chapter 5

Conclusion

This report presents a more accessible approach to implementing a Machine Translation (MT) system based on Large Language Models (LLMs), specifically tailored for small and medium-sized enterprises (SMEs). By adhering to guidelines from leading AI companies with deep expertise in LLM development, we successfully designed and implemented the system, demonstrating its effectiveness and versatility in a real-world scenario.

The results confirm that it is feasible to develop a robust, state-of-the-art LLM-based MT system without relying on external services provided by larger companies. This autonomy has significant implications for the industry, as it empowers SMEs to be more competitive while potentially reducing the technological gap between them and larger corporations.

5.1 Limitations

Several limitations became evident throughout the development process. One of the most significant challenges was the lack of access to our own GPUs, which made model training more expensive, as we had to rely on third-party services such as Google Colab. This increased the overall cost of experiments and limited the scalability of our efforts. Another notable limitation was the system's performance on low-resource languages, such as Slovene

and Hungarian, which demonstrated lower accuracy compared to more commonly tackled languages within the company. This highlighted the system's difficulty in handling languages with limited available data.

Scaling multilingual models, such as mBart and mT5, also presented challenges. Unlike single-language models, where updates can be isolated to specific language pairs, multilingual models require updating all weights simultaneously, increasing complexity and time demands. Additionally, the need to self-host the models further added to resource requirements. Reproducibility, a critical challenge in large language model development, also demands substantial time and effort to ensure consistent results, and when combined with the slower time-to-market, these factors limit the system's practical application and competitiveness.

5.2 Future work

While this work demonstrated the potential of LLMs for Machine Translation systems in SMEs, there are many opportunities for further improvement. Optimizing hardware resources is a key area for future work. Exploring alternatives such as cloud-based GPU rental services or investing in dedicated in-house hardware could significantly reduce costs and remove hardware restrictions, enhancing scalability.

Improving support for low-resource languages is another priority for future research. Expanding the system to cover additional languages requires solutions for these low-resource contexts. Techniques like data augmentation or the use of pivot languages for translation [20] could be explored to address this issue. These strategies, along with continued refinements to model scalability, efficiency, and reproducibility, will be crucial for advancing the application of LLM-based systems in SMEs.

Bibliography

- [1] S. Alansary. A formalized reference grammar for UNL-based machine translation between English and Arabic. In M. Kay and C. Boitet, editors, *Proceedings of COLING 2012: Posters*, pages 33–42, Mumbai, India. The COLING 2012 Organizing Committee, December 2012. URL: <https://aclanthology.org/C12-2004>.
- [2] A. S. Ariesandy, M. Amien, A. F. Aji, and R. E. Prasajo. Synthetic source language augmentation for colloquial neural machine translation, 2020. arXiv: 2012.15178 [cs.CL]. URL: <https://arxiv.org/abs/2012.15178>.
- [3] Y. M. Asano, C. Rupprecht, and A. Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image, 2020. arXiv: 1904.13132 [cs.CV]. URL: <https://arxiv.org/abs/1904.13132>.
- [4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2016. arXiv: 1409.0473 [cs.CL]. URL: <https://arxiv.org/abs/1409.0473>.
- [5] B. Bickel and J. Nichols. Fusion of selected inflectional formatives (v2020.3). In M. S. Dryer and M. Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo, 2013. DOI: 10.5281/zenodo.7385533. URL: <https://doi.org/10.5281/zenodo.7385533>.

- [6] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. URL: <https://aclanthology.org/J90-2002>.
- [7] D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007. DOI: 10.1162/coli.2007.33.2.201. URL: <https://aclanthology.org/J07-2003>.
- [8] A. M. Dai and Q. V. Le. Semi-supervised sequence learning, 2015. arXiv: 1511.01432 [cs.LG]. URL: <https://arxiv.org/abs/1511.01432>.
- [9] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. arXiv: 1810.04805. URL: <http://arxiv.org/abs/1810.04805>.
- [10] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. arXiv: 1810.04805. URL: <http://arxiv.org/abs/1810.04805>.
- [11] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn. An attentional model for speech translation without transcription. In K. Knight, A. Nenkova, and O. Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 949–959, San Diego, California. Association for Computational Linguistics, June 2016. DOI: 10.18653/v1/N16-1109. URL: <https://aclanthology.org/N16-1109>.
- [12] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, E. Grave, M. Auli, and A. Joulin. Beyond

- english-centric multilingual machine translation, 2020. arXiv: 2010.11125 [cs.CL]. URL: <https://arxiv.org/abs/2010.11125>.
- [13] M. Fernández-Rodríguez. Evolución de la traducción asistida por ordenador. de las herramientas de apoyo a las memorias de traducción. *Sendebarr*, 21:201–230, dic. 2010. DOI: 10.30827/sendebarr.v21i0.374. URL: <https://revistaseug.ugr.es/index.php/sendebarr/article/view/374>.
- [14] M. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O’Regan, S. Ortiz-Rojas, J. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25:127–144, June 2011. DOI: 10.1007/s10590-011-9090-0.
- [15] M. Galley, M. Hopkins, K. Knight, and D. Marcu. What’s in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 273–280, Boston, Massachusetts, USA. Association for Computational Linguistics, May 2004. URL: <https://aclanthology.org/N04-1035>.
- [16] B. Haddow, R. Bawden, A. V. Miceli Barone, J. Helcl, and A. Birch. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732, September 2022. DOI: 10.1162/coli_a_00446. URL: <https://aclanthology.org/2022.c1-3.6>.
- [17] J. Howard and S. Ruder. Universal language model fine-tuning for text classification, 2018. arXiv: 1801.06146 [cs.CL]. URL: <https://arxiv.org/abs/1801.06146>.
- [18] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2024.

- URL: <https://web.stanford.edu/~jurafsky/slp3/>. Online manuscript released August 20, 2024.
- [19] D. Kenny. *Machine Translation for everyone: Empowering users in the age of artificial intelligence*. Berling: Language Science Press, 2022.
- [20] Y. Kim, P. Petrov, P. Petrushkov, S. Khadivi, and H. Ney. Pivot-based transfer learning for neural machine translation between non-English languages. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 866–876, Hong Kong, China. Association for Computational Linguistics, November 2019. DOI: 10.18653/v1/D19-1080. URL: <https://aclanthology.org/D19-1080>.
- [21] D. L. King. The impact of multilingualism on global education and language learning. *Cambridge Assessment English*, 2018.
- [22] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL: <https://aclanthology.org/N03-1017>.
- [23] C. N. Li and S. A. Thomson. *Mandarin Chinese. A Functional Reference Grammar*. University of California Press, 1981.
- [24] C.-Y. Lin. ROUGE: a package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics, July 2004. URL: <https://aclanthology.org/W04-1013>.

- [25] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation, 2020. arXiv: 2001.08210 [cs.CL]. URL: <https://arxiv.org/abs/2001.08210>.
- [26] M.-T. Luong and C. D. Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Association for Computational Linguistics (ACL)*, Berlin, Germany, August 2016. URL: https://nlp.stanford.edu/pubs/luong2016acl_hybrid.pdf.
- [27] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics, September 2015. URL: <http://aclweb.org/anthology/D15-1166>.
- [28] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 133–139, USA. Association for Computational Linguistics, 2002. DOI: 10.3115/1118693.1118711. URL: <https://doi.org/10.3115/1118693.1118711>.
- [29] J. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549, 2006. DOI: 10.1162/coli.2006.32.4.527. URL: <https://aclanthology.org/J06-4004>.
- [30] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: a survey, 2024. arXiv: 2402.06196 [cs.CL]. URL: <https://arxiv.org/abs/2402.06196>.

- [31] A. Mohammadshahi, V. Nikoulina, A. Berard, C. Brun, J. Henderson, and L. Besacier. SMaLL-100: introducing shallow multilingual machine translation model for low-resource languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8348–8359, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics, December 2022. URL: <https://aclanthology.org/2022.emnlp-main.571>.
- [32] A. Mohammadshahi, V. Nikoulina, A. Berard, C. Brun, J. Henderson, and L. Besacier. What do compressed multilingual machine translation models forget? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4308–4329, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics, December 2022. URL: <https://aclanthology.org/2022.findings-emnlp.317>.
- [33] M. Neff and M. McCord. Acquiring lexical data from machine-readable dictionary resources for machine translation. In 1990.
- [34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In P. Isabelle, E. Charniak, and D. Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics, July 2002. DOI: 10.3115/1073083.1073135. URL: <https://aclanthology.org/P02-1040>.
- [35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In M. Walker, H. Ji, and A. Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*,

- pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics, June 2018. DOI: 10.18653/v1/N18-1202. URL: <https://aclanthology.org/N18-1202>.
- [36] T. Poibeau. *Machine Translation*. The MIT Press, 2017.
- [37] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. In 2018.
- [38] S. Ranathunga, E.-S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam, and R. Kaur. Neural machine translation for low-resource languages: a survey, 2021. arXiv: 2106.15115 [cs.CL]. URL: <https://arxiv.org/abs/2106.15115>.
- [39] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958. URL: <https://api.semanticscholar.org/CorpusID:12781225>.
- [40] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.
- [41] J. Shin, Y. Lee, S. Yoon, and K. Jung. Fast and accurate deep bidirectional language representations for unsupervised learning. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 823–835, Online. Association for Computational Linguistics, July 2020. DOI: 10.18653/v1/2020.acl-main.76. URL: <https://aclanthology.org/2020.acl-main.76>.
- [42] R. Shyam and R. Chakraborty. Machine learning and its dominant paradigms. 8:2021, September 2021. DOI: 10.37591/JoARB.

- [43] J. Slocum. A survey of machine translation: its history, current status and future prospects. *Computational Linguistics*, 11(1):1–17, 1985. URL: <https://aclanthology.org/J85-1001>.
- [44] X. Tan, J. Chen, D. He, Y. Xia, T. Qin, and T.-Y. Liu. Multilingual neural machine translation with language clustering. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics, November 2019. DOI: 10.18653/v1/D19-1089. URL: <https://aclanthology.org/D19-1089>.
- [45] J. Tiedemann. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, Y. Graham, P. Guzman, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, A. Martins, M. Morishita, C. Monz, M. Nagata, T. Nakazawa, and M. Negri, editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics, November 2020. URL: <https://aclanthology.org/2020.wmt-1.139>.
- [46] J. Tiedemann, M. Aulamo, D. Bakshandaeva, M. Boggia, S.-A. Grönroos, T. Nieminen, A. Raganato, Y. Scherrer, R. Vazquez, and S. Virpioja. Democratizing neural machine translation with OPUS-MT. *Language Resources and Evaluation*, (58):713–755, 2023. ISSN: 1574-0218. DOI: 10.1007/s10579-023-09704-w.
- [47] J. Tiedemann and S. Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

- [48] D. Torregrosa, N. Pasricha, M. Masoud, B. R. Chakravarthi, J. Alonso, N. Casas, and M. Arcan. Leveraging rule-based machine translation knowledge for under-resourced neural machine translation models. In M. Forcada, A. Way, J. Tinsley, D. Shterionov, C. Rico, and F. Gaspari, editors, *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pages 125–133, Dublin, Ireland. European Association for Machine Translation, August 2019. URL: <https://aclanthology.org/W19-6725>.
- [49] H. Tyagi, P. Jung, and H. Lee. Machine translation to control formality features in the target language, 2023. arXiv: 2311.13475 [cs.CL]. URL: <https://arxiv.org/abs/2311.13475>.
- [50] E. Union. Eur-lex. 1998-2024. URL: <https://eur-lex.europa.eu>.
- [51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. arXiv: 1706.03762. URL: <http://arxiv.org/abs/1706.03762>.
- [52] S. Vogel, H. Ney, and C. Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2, COLING '96*, pages 836–841, Copenhagen, Denmark. Association for Computational Linguistics, 1996. DOI: 10.3115/993268.993313. URL: <https://doi.org/10.3115/993268.993313>.
- [53] T. von der Brück and M. Pouly. Text similarity estimation based on word embeddings and matrix norms for targeted marketing. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1827–1836, Minneapolis, Minnesota. Association for

- Computational Linguistics, June 2019. DOI: 10 . 18653 / v1 / N19 - 1181. URL: <https://aclanthology.org/N19-1181>.
- [54] L. Wang, Z. Tu, A. Way, and Q. Liu. Exploiting cross-sentence context for neural machine translation. In M. Palmer, R. Hwa, and S. Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics, September 2017. DOI: 10 . 18653 / v1 / D17 - 1301. URL: <https://aclanthology.org/D17-1301>.
- [55] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google’s neural machine translation system: bridging the gap between human and machine translation, 2016. arXiv: 1609 . 08144 [cs.CL]. URL: <https://arxiv.org/abs/1609.08144>.
- [56] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. Mt5: a massively multilingual pre-trained text-to-text transformer, 2021. arXiv: 2010 . 11934 [cs.CL]. URL: <https://arxiv.org/abs/2010.11934>.
- [57] K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France. Association for Computational Linguistics, July 2001. DOI: 10 . 3115 / 1073012 . 1073079. URL: <https://aclanthology.org/P01-1067>.
- [58] S. Yang, Y. Wang, and X. Chu. A survey of deep learning techniques for neural machine translation, 2020. arXiv: 2002 . 07526 [cs.CL]. URL: <https://arxiv.org/abs/2002.07526>.

-
- [59] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models, 2023. arXiv: 2303.18223 [cs.CL]. URL: <https://arxiv.org/abs/2303.18223>.
- [60] W. Zhu, H. Liu, Q. Dong, J. Xu, S. Huang, L. Kong, J. Chen, and L. Li. Multilingual machine translation with large language models: empirical results and analysis, 2023. arXiv: 2304.04675 [cs.CL].

Acknowledgements

I would like to express my deep gratitude to the Medhiartis team for giving me the opportunity to contribute to this project. I am especially thankful to my project manager, Sara Rossetti, and my colleague, Rooshan Saleem Butt, for their unwavering support throughout the entire process and for providing everything I needed to succeed. I also extend my sincere thanks to Jessica Bottaro for the beautiful designs she created for our web application, and to Federica Nigro for her excellent coordination of feedback from the translation team.

I am equally grateful to my supervisor, Paolo Torrioni, and my co-supervisor, Gianmarco Pappacoda, for their guidance and invaluable assistance during the writing process.