UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
BACHELOR IN COMPUTER SCIENCE

ALEX SANTOS DE OLIVEIRA
RAFAEL DA SILVA FERNANDES

EXPLORING THE IMPACT OF INTERMEDIATE LANGUAGES ON MACHINE TRANSLATION

RIO DE JANEIRO
2023

ALEX SANTOS DE OLIVEIRA
RAFAEL DA SILVA FERNANDES

EXPLORING THE IMPACT OF INTERMEDIATE LANGUAGES ON MACHINE TRANSLATION

Undergraduate dissertation submitted to the Instituto de Computação, Universidade Federal do Rio de Janeiro as a partial requirement to obtain the title of Bachelor in Computer Science.

Orientador: Prof. João Antonio Recio da Paixão
Laura de Oliveira Fernandes Moraes

RIO DE JANEIRO
2023

Oliveira, Alex Santos de

O48e Exploring the impact of intermediate languages on machine translation / Alex Santos de Oliveira e Rafael da Silva Fernandes. – 2023.

66 f.

Orientador: João Antonio Recio da Paixão.
Coorientadora: Laura de Oliveira Fernandes Moraes.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2023.

1. Tradução por máquina. 2. Word Embedding. 3. Problema Ortogonal de Procrustes. 4. Língua intermediária. I. Fernandes, Rafael da Silva. II. Paixão, João Antonio Recio da (Orient.). III. Moraes, Laura de Oliveira Fernandes (Coorient). IV. Universidade Federal do Rio de Janeiro, Instituto de Computação. V. Título.

ALEX SANTOS DE OLIVEIRA
RAFAEL DA SILVA FERNANDES

EXPLORING THE IMPACT OF INTERMEDIATE LANGUAGES ON MACHINE TRANSLATION

Undergraduate dissertation submitted to the Instituto de Computação, Universidade Federal do Rio de Janeiro as a partial requirement to obtain the title of Bachelor in Computer Science.

Approved in Rio de Janeiro, September 5th, 2023

João Antonio Recio da Paixão, UFRJ
Supervisor

Laura de Oliveira F. Moraes

Laura de Oliveira Fernandes Moraes,
UNIRIO
Supervisor

Daniel Sadoc Menasché

Daniel Sadoc Menasché, UFRJ
Examiner

Felipe Fink Grael, Twist Systems
Examiner

João Vitor de Oliveira Silva

João Vitor de Oliveira Silva, University of
Colorado - Denver
Examiner

ACKNOWLEDGEMENTS

We extend our heartfelt gratitude to Professor João Antonio Recio da Paixão, whose invaluable guidance played a pivotal role in directing and enabling the successful execution of this project. His expertise and support were instrumental in shaping our research and making it feasible.

We would also like to express our sincere appreciation to Laura de Oliveira Fernandes Moraes, our co-advisor, for her valuable contributions and insightful feedbacks throughout this project. Her guidance and encouragement greatly enriched our work.

Additionally, we are thankful to João Vitor de Oliveira Silva for providing the initial feedback on our project. His constructive input and thoughtful suggestions have significantly contributed to the refinement of our study.

"Facts which at first seem improbable will, even on scant explanation, drop the cloak which has hidden them and stand forth in naked and simple beauty."

Galileo Galilei

RESUMO

A área de tradução é mais antiga que o computador e, conforme a tecnologia foi avançando, ela foi se modernizando e se adaptando às novas descobertas, tentando sempre se tornar mais eficiente e precisa. Tradução por máquina, uma parte integral do Processamento de Linguagem Natural, procura possibilitar a tradução automática entre idiomas buscando sempre melhorar a precisão e a acessibilidade. Porém, considerando a quantidade de idiomas que existem, treinar modelos para todos os pares de idiomas possíveis sem o uso de múltiplos computadores poderosos e uma quantidade imensa de dados se torna uma tarefa complexa, além de ser impossível para alguns pares de idiomas. Neste trabalho nós avaliamos uma forma simples e rápida de diminuir o número de treinamentos e como ela impacta na qualidade da tradução. Nossos resultados mostraram que é possível realizar traduções usando idiomas intermediários ao invés de se traduzir diretamente para o idioma desejado sem impactar de forma significativa no resultado da tradução. Também mostramos que o impacto está relacionado com a família dos idiomas original, alvo, e intermediário. Com isso, concluímos que usar idiomas intermediários é uma técnica efetiva para diminuir de forma o número de treinamentos necessários ao se lidar com um número grande de idiomas, fazendo com que o processo de treinar modelos demande menos recursos. Isso permite que a criação de modelos para tradução usando múltiplos idiomas se torne mais acessível já que, por exemplo, ao usar 10 idiomas, treinar uma tradução direta entre todos os pares possíveis de idiomas resultaria em 45 treinamentos diferentes, número obtido calculando a combinação simples. Já usando um idioma intermediário para esse mesmo cenário, só seria necessário calcular traduções para esse idioma, resultando em apenas 9 treinamentos, reduzindo o custo computacional, além de beneficiar diversas áreas e beneficiar a troca de conhecimentos e ideias.

Palavras-chave: Tradução por Máquina; Word Embedding; Problema Ortogonal de Procrustes; Língua Intermediária

ABSTRACT

The field of translation predates the computer, and as technology has advanced, it has evolved and adapted to new discoveries, constantly striving to become more efficient and precise. Machine translation, an integral part of Natural Language Processing, seeks to enable automatic translation between languages with the ongoing pursuit of enhanced accuracy and accessibility. However, considering the vast number of languages in existence, training models for all possible language pairs without the use of powerful computers and massive amounts of data becomes a complex task, and it is infeasible for certain language pairs. In this study, we evaluate a simple and efficient approach to reduce the number of training instances and its impact on translation quality. Our results demonstrate that it is possible to perform translations using intermediate languages instead of translating directly to the desired language without significantly impacting the translation outcome. We also show that the impact is related to the language family of the source, target, and intermediate languages. Hence, we conclude that using intermediate languages is an effective technique for reducing the number of required training instances when dealing with a large number of languages, making the model training process more resource-efficient. This approach enables the creation of translation models using multiple languages to become more accessible. For instance, when using 10 languages, training direct translations between all possible language pairs would require 45 distinct training instances, calculated using the combination formula. However, by employing an intermediate language in this scenario, only translations to and from that language would need to be computed, resulting in just 9 training instances. This reduction in computational cost not only benefits various fields but also fosters the exchange of knowledge and ideas across diverse communities.

Keywords: Machine Translation; Word Embedding; Orthogonal Procrustes Problem; Intermediary Language

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ASCII	American Standard Code for Information Interchange
BLEU	Bilingual Evaluation Understudy
CBOW	Continuous Bag of Words
ML	Machine Learning
MT	Machine Translation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
SMT	Statistical Machine Translation
SVD	Singular Value Decomposition

CONTENTS

1 Introduction

Translations play a crucial role in numerous aspects of modern life, offering significant value and importance across various settings. In academia, translations enable researchers to access and explore foreign articles, expanding their understanding and knowledge in specific subjects (MUNDAY, 2010). In journalism, translations provide a valuable tool for gaining diverse perspectives on complex global issues, allowing journalists to present a comprehensive view to their audiences (DOORSLAER, 2010).

Regarding the academic and research domains, translation facilitates the exchange and sharing of knowledge across borders. While the majority of scientific researches are conducted and published in English [1], many research groups and laboratories speak other languages. By translating a research, it is possible to make a discovery or theory available to a wider audience who is able to build upon and expand that research even further (TITLER, 2018).

As a response to the increasing demand for efficient and scalable translation solutions, Machine Translation (MT) has emerged as a valuable tool in the language industry. MT systems utilize advanced algorithms and language models to automatically translate texts, providing a faster alternative to traditional human translation processes (KOEHN, 2009). One of the key advantages of MT is its ability to handle large volumes of content and quickly deliver translations, making it particularly suitable for time-sensitive materials, e.g. internal communications or chat-bots, where quick turnaround is prioritized over absolute translation quality (JANSSENS; LAMBERT; STEYAERT, 2004).

An associated concept in the realm of translation is that of indirect translation. Unlike direct translation, which occurs directly between two languages, indirect translation involves the insertion of an intermediate language in between, where the text is first translated into the intermediate language and then into the target language. It suggests that leveraging an intermediary language can potentially address some of the challenges and demands posed by MT. In cases where a direct translation is not feasible due to resource constraints or the lack of existing language pairs, utilizing an intermediary language might serve as a viable solution to bridge the translation gap (COHN; LAPATA, 2007).

In addition to facilitating the translation process itself, intermediary languages can also be useful for other tasks in Natural Language Processing (NLP). For example, (ZHAO; SARKAR, 2011) wrote they can be used to identify similarities and differences between languages, to extract key features or patterns in language usage, or to improve Machine Learning (ML) algorithms for language-related tasks.

[1] https://www.theatlantic.com/science/archive/2015/08/english-universal-language-science-research/400919/

1.1 Motivation

Natural language translation in a traditional way requires an one-on-one translation between two languages, requiring translators who are proficient in both the languages. However, with the growing need for multilingual communication (GROUP, 2018), the demand for translating a language into multiple other languages and vice versa has increased significantly. Developing translation packages for each language pair can be a time-consuming and costly process, taking approximately 2 to 3 years and requiring substantial effort (SMITH, 2019).

Even if we argue in favor of utilizing MT to address the challenges of traditional one-on-one translation, it's important to acknowledge that this automated approach brings its own set of demands. MT heavily relies on substantial amounts of data for training, especially when aiming for high translation accuracy. This reliance on extensive datasets, along with the computational resources required for training complex models, can lead to high computational costs (KENNY, 2018).

Motivated by the pursuit of a more streamlined solution, we embarked on an exploration into the concept of intermediary languages (COHN; LAPATA, 2007). These languages offer reusability and potential reductions in time and data requirements.

The significance of intermediary languages extends beyond the realm of translation and has held a crucial position within the field of computer science for quite some time. Numerous authors, including Barendregt (BARENDREGT et al., 1987), Hardwick (HARDWICK; SIPELSTEIN, 1996), and Denker (DENKER; MILLEN, 1999), have highlighted their essential role. Intermediary languages have been frequently employed to facilitate seamless communication between disparate systems, enabling the efficient exchange of information across different platforms. This characteristic has proven to be particularly advantageous in computer science, where interoperability and data interchangeability are paramount.

Moreover, the utilization of intermediary languages also presents the advantage of reducing the complexity associated with combinations. In scenarios involving multiple languages, the direct translation between all possible language pairs would necessitate a considerable number of training instances, as determined by the combination formula. By introducing an intermediate language, the need for direct translations between all pairs is circumvented, and translations are only required to and from the intermediate language.

While the utilization of intermediary languages offers the promise of alleviating some of the complexities inherent in MT efforts, it is essential to recognize that this approach is not without its repercussions. The central aim of this study is to examine the impact of employing intermediary languages on the translation process. By introducing an additional step of translation through an intermediary language, we intend to evaluate whether this method introduces potential drawbacks that could compromise the translation quality and accuracy achieved through direct translations.

1.2 Objective

In our research, we aim to investigate the extent of the impact of indirect translation involving intermediary languages before reaching the final target language. To conduct this investigation, we explore contexts where the technique becomes applicable, particularly considering the availability of sufficient multilingual data that allows us to perform both direct and indirect translations.

It is particularly crucial to ascertain the reliability of such translations in scenarios where intermediary languages are employed. To address these questions, we have chosen languages from two distinct language families. The first family belongs to the Romance group and includes Portuguese, Spanish, and Italian. The second family encompasses languages of Anglo-Saxon origin, such as English, German, and Swedish. In this endeavor, our primary goal is to compare the performance of direct translations with that of indirect translations.

After conducting direct translations of sentences between languages from both the same and different language families, we proceeded with the evaluation process. In this evaluation, we assessed all possible translation scenarios involving an intermediate language. These scenarios encompassed translations using languages solely from the same family (e.g., Portuguese to Italian to Spanish), translations with single language family change (e.g., Portuguese to Spanish to English), and translations with double language family change (e.g., Portuguese to English to Spanish).

Figure 1 depicts a graphical representation of the translation paths involving multiple languages in our study. In this graph, each vertex corresponds to a specific language, and the edges represent the translation paths between languages. For instance, if we have the sequence "Portuguese → English → Spanish", it signifies that we first translate from Portuguese to English and then from English to Spanish. This graph provides a visual overview of the various translation scenarios explored, showcasing the connections between different languages and the utilization of intermediate languages in the translation process.

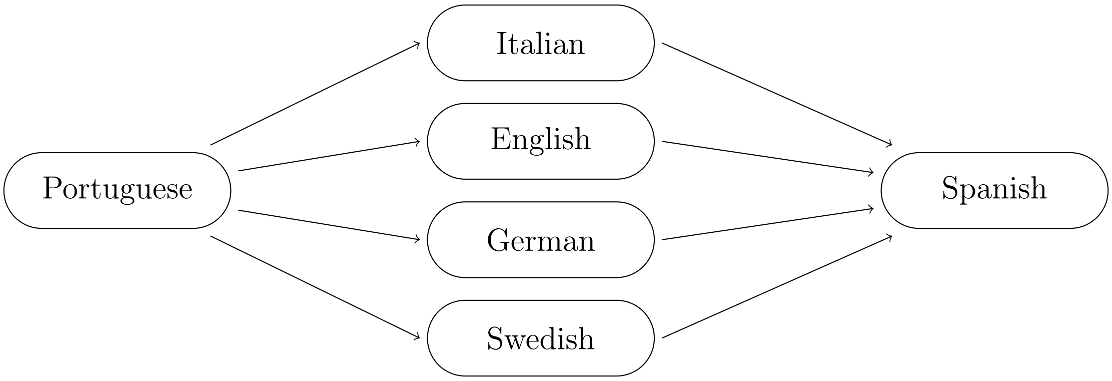

Figure 1 – Multiple options to translate from Portuguese to Spanish using an intermediary language

1.3 Approach formulation

We propose a translation approach that involves both direct translation and translation utilizing an intermediary language. This approach aims to evaluate the impact of intermediary language translation on the quality of the final translation output. By comparing the outcomes of both direct and indirect translation processes, we intend to understand the benefits and potential drawbacks of employing an intermediary language.

Our proposed approach involves a series of key steps to generate the final translated output. Initially, the words in the sentences are represented as word embeddings, which capture their semantic meanings within a high-dimensional vector space. Subsequently, the vector summation of the word embeddings is performed to calculate the vector of each sentence. Finally, the translation is made by aligning source and target languages using the orthogonal Procrustes problem, a technique that optimizes the transformation between the two vector spaces. This alignment ensures that corresponding sentences in the different languages share similar positions in the vector space, enabling effective translation.

It is important to note that our approach is grounded in the fundamental assumption that, in a word embedding, words with similar meanings occupy proximate regions within the vector space. Consequently, when the vectors of words from a sentence are summed, their resultant vector is expected to point in a direction that represents its meaning that, when aligned with other languages, will be pointing to a similar direction of its equivalent.

With this conceptual foundation, we proceed to evaluate the effectiveness of our approach. This evaluation is facilitated through the calculation of cosine similarity, a metric commonly employed in measuring the similarity between vectors. In the context of word embeddings, cosine similarity assesses the proximity of words based on the directions in which their vectors point, enabling us to gauge their semantic similarity. Additionally, we utilize two other evaluation metrics: BLEU, a widely recognized measure in translation tasks (PAPINENI et al., 2002); and Euclidean distance, which provides an absolute measure of the distance between vectors in the vector space. These metrics collectively allow us to assess the quality and accuracy of our translations, taking into account both relative and absolute measures.

Following the approach formulation, the next chapters of this work will explore the related works, theoretical foundation, methodology, results, and conclusion. These chapters aim to provide a better understanding of the research process and the outcomes obtained.

- **Chapter 2 - Related Works:** this chapter provides an overview of relevant studies and research conducted in the field of translations, using or not intermediate languages. It examines previous works that have explored similar topics, methodologies, and approaches.
- **Chapter 3 - Theoretical Foundation:** this chapter establishes the theoretical framework that underpins the research on translations using intermediate languages. It explores relevant concepts, theories, and previous studies that inform the investigation.
- **Chapter 4 - Methodology:** in this chapter, the research methodology is described in detail. It outlines the steps taken to collect and process the translation dataset, as well as the mathematical models and techniques employed for analysis. The chapter explains the rationale behind the chosen methodology, highlighting its suitability for addressing the research objectives.
- **Chapter 5 - Results:** this chapter presents the findings derived from the analysis of translations using intermediate languages. The results are organized and presented in a manner that facilitates interpretation and understanding.
- **Chapter 6 - Conclusion:** the conclusion chapter summarizes the key findings of the research and discusses their implications. It revisits the research objectives and evaluates the extent to which they have been achieved. The chapter acknowledges the limitations of the study and proposes avenues for future research.

2 Related works

2.1 Machine Translation

Machine Translation (MT) is a task that seeks to enable automatic translation between different languages. This field of study has been a subject of research for over 70 years, evolving in tandem with the development of computers. Notably, one of the big advancement and most used techniques recently was the creation of Neural Machine Translation (NMT) (WANG et al., 2021).

NMT is an approach towards automated translation using Machine Learning (ML) to translate texts from one language into another. A division of computational linguistics, NMT relies on artificial neural networks (DREW; MONSON, 2000) to predict the likelihood of certain sequences of words. The NMT algorithm is an example of Deep Learning: users can train NMT engines to recognise source and target connections using large datasets. As connections between words are strengthened or weakened through training on the datasets, the machine observes these correlations and adapts to predict and increase the likelihood of correct translations (STAHLBERG, 2020).

In a study conducted by (YANG; OGATA, 2019), the authors employed a NMT approach to perform translation between English and Japanese. The authors reported that training the model took approximately 27 hours for each experiment using 4 GPUs, and the translation process achieved high-quality results with competitive performance in terms of both accuracy and fluency.

Another notable study by (CHEN et al., 2016) focused on translation between Chinese and English. The translation experiments were conducted on a large-scale dataset, and the authors reported that the translation process required considerable computational resources, taking several hours to complete on a high-performance computing cluster. The results demonstrated significant improvements in translation quality compared to traditional Statistical Machine Translation (SMT) approaches.

In a different approach, (ZENS; OCH; NEY, 2002) investigated the use of phrase-based SMT for English to German translation. The authors utilized parallel *corpora* to train their translation model. The training process involved several steps, including data cleaning, tokenization, and alignment, and it took several days to complete due to the large-scale nature of the dataset. The authors reported competitive translation quality with promising results.

2.2 Aligning embeddings

A method to represent words in a way the computer is able to manipulate them and perform the necessary calculations is required. For that reason, we are using "word embeddings" (YIN; SHEN, 2018), which are representations of words as a vector of numbers. Since we are using word embeddings for cross-lingual translation, it is important to note that one word embedding per language is required.

Aligning two or more word embeddings to perform translations is an idea that has already been explored in other studies. In (MIKOLOV; LE; SUTSKEVER, 2013), the researchers addressed the same optimization problem. They utilized Word2Vec as the algorithm to create the embeddings and Google Translator to compile a list of known translations.

In (XING et al., 2015), the authors built upon the work mentioned in the previous paragraph and proposed an alternative method to align the embeddings. They introduced a technique that involves normalizing the word vectors on a hyper-sphere and imposing constraints on the linear transform as an orthogonal transform. This approach aimed to further enhance the alignment and translation capabilities of word embeddings.

In (QI et al., 2018), the authors pre-trained word embeddings to help in NMT tasks. Since NMT tasks often suffer in low-resource scenarios where sufficiently large-space parallel *corpora* be obtained, their experiments proved that pre-trained word embeddings are invaluable for improving performance. They also compare the effect of language similarity, where the main intuitive hypothesis as to why pre-training works is that the embedding space becomes more consistent, with semantically similar words closer to each other. The authors make an additional hypothesis: if the two languages in the translation pair are more linguistically similar, the semantic neighborhoods will be more similar between the two languages, i.e., semantic distinctions, or "polysemy", will likely manifest themselves in more similar ways across more similar languages.

In another study, by (GRAVE; JOULIN; BERTHET, 2019), the researchers employed a variation of the orthogonal Procrustes problem to address the alignment of word embeddings. Unlike the previous approaches, this method was designed as an unsupervised technique, meaning it did not rely on any pre-existing bilingual information. Instead, the researchers aimed to refine the alignment iteratively, continuously improving the alignment without the need for known bilingual pairs.

According to (ZOU et al., 2013), bilingual word embeddings are introduced - which are semantic embeddings associated across two languages - in the context of NMT. The bilingual embeddings capture not only semantic information of monolingual words, but also semantic relationships across different languages. This property allows them to define semantic similarity metrics across phrase-pairs, which makes them very important for machine translation tasks. Specifically in this paper, the authors method uses word alignment to learn bilingual embeddings.

In (DONANDT; CHIARCOS, 2019), the authors construct a multi-lingual word embedding space by projecting new languages in the feature space of a language for which a pre-trained embedding model exists. They use the similarity of the word embeddings to predict candidate translations. Their contribution is based on the application of a technology originally developed for a related, but broader problem, the identification of cognates in dictionaries of languages that are either diachronically or culturally related with each other (ARNAUD et al., 2017). Cognate candidates can be identified by means of phonological and semantic similarity metrics, and the latter are the basis for the implementation that they describe with this paper.

In their study, (VYAS; CARPUAT, 2016) an innovative approach to ascertain the meaning of a word in one language by using words from another language. They achieve this by utilizing sparse non-negative embeddings, a method that represents word contexts with each dimension having an interpretable meaning. Subsequently, the authors align these word representations, resulting in a sparse bilingual word representation with interpretable dimensions. To evaluate their approach, they create a test set for English to French translations with the assistance of crowdsourcing [2]. Remarkably, their method achieved an impressive 70% F1-accuracy on cross-lingual lexical entailment tasks, showcasing the potential effectiveness of their approach in understanding word meanings across different languages.

In their investigation (UPADHYAY et al., 2016), the researchers carried out an empirical comparison of four cross-lingual methods that vary in the level of supervision required. The evaluation involved testing these methods across multiple tasks, including monolingual word similarity, cross-lingual dictionary induction, cross-lingual document classification, and cross-lingual syntactic dependency parsing. Surprisingly, their findings revealed an interesting trend: models that require more bilingual knowledge tend to yield better results. However, they also observed that models with access to less data could still perform exceptionally well, depending on the specific task at hand, and even compete with more computationally expensive approaches.

2.3 Comparison

It is important to highlight that our work differs from existing studies in several key aspects. Firstly, unlike previous research, we utilize a unique and specific dataset for our translation task. It should be noted that, to the best of our knowledge, no prior study has utilized the exact dataset we employ in our research. Therefore, a direct comparison with previous works in terms of translation performance may not be possible.

[2] https://dictionary.cambridge.org/us/dictionary/english/crowdsourcing

Furthermore, our work distinguishes itself from existing studies by employing a unique combination of techniques: word embeddings and intermediary languages. Unlike previous research, we leverage word embeddings to capture the contextual information encoded in the embeddings. Additionally, the inclusion of intermediary languages enables us to explore the advantages and challenges associated with indirect translations.

As it was not feasible to directly perform a comparison with these referenced studies in terms of translation runtime, we have instead referred to them to present the technologies and techniques employed in translation research. These studies have reported runtimes for their respective translation processes, considering factors such as dataset size, model complexity, and computational resources available. While we cannot directly apply their reported runtimes to our specific implementation, we have considered these references to inform our research and methodology.

3 Theoretical Foundation

In this study, we aim to ensure that readers can comprehend the topics covered by establishing and clarifying essential fundamental concepts. We recognize that a basic understanding of linear algebra is necessary for a better understanding of the material presented.

As we present the subsequent sections, we will explore these concepts in detail and provide explanations, making the content accessible and easy to comprehend for all readers. Our goal is to offer clear and straightforward explanations of the underlying principles, fostering a deeper understanding of the subject matter.

3.1 Embedding

This study focuses on data that consists of text, or textual type data. When represented in a computer, this data is typically in the form of strings, which are sequences of characters. While some ML models require numerical data, string data is often not sufficient for many purposes. In this particular study, the model being used requires numerical input, so it is necessary to convert the original string data into a numerical format.

3.1.1 Character level

One common method for converting text into numerical data is to substitute each character with a corresponding number. This approach is possible because each character in a computer's memory is already represented by a specific numerical value. However, for users and in programming languages, characters are typically treated in the familiar way, without explicitly showing their numerical representation.

Computers use an encoding system to represent characters, which specifies which memory value corresponds to each glyph or symbol. For example, the American Standard Code for Information Interchange (ASCII), described in (TABLE, 1979), assigns the value 65 to the character "A", 66 to "B", and so on. To convert text into numerical sequences, we can use the ASCII code and substitute each character with its corresponding numerical value. Using this approach, the text "Hello World" would be encoded as a sequence of numbers as follows:

H	e	l	l	o		W	o	r	l	d
72	101	108	108	111	32	87	111	114	108	100

While this method is simple and widely used, it has limitations due to many numbers not having a corresponding glyph, and it cannot support a variety of characters. Unicode (DAVIS; COLLINS, 1990) is a more comprehensive encoding, but there is an easier alternative.

To determine the optimal encoding, it is important to first specify the requirements of the solution. In this case, the aim is to create a mapping between each character in the dataset, or *corpus*, of authentic text and a corresponding numerical representation. This is achieved by generating a list of all the characters present in the text, and assigning each character a numerical index. This method ensures that there is a precise match between the characters used in the dataset and their numerical representation, with no extraneous values.

However, there is an inherent issue with this approach. By assigning each character a numerical value, an implicit ordering is created. Additionally, each character now has a specific magnitude associated with it. For instance, if “B” is assigned the value 2 and “J” is assigned the value 10, it is established that “B” comes after “J” in the ordering and that “B” is 5 times smaller than ´‘J”. From a semantic perspective, such a relationship is meaningless.

3.2 One-hot encoding

One way to circumvent the adversity mentioned above is to designate different dimensions to represent each character. In this encoding, known as “one-hot encoding” (CERDA; VAROQUAUX; KÉGL, 2018), each character is represented by a multidimensional vector. In this vector, each dimension corresponds to one character. Therefore, if only the letters of the Latin alphabet are used, the dimension of the vector would be 26. One simply needs to put the value corresponding to the dimension of this character as 1 and all others as 0, as shown in Figure 2, to encode a character.

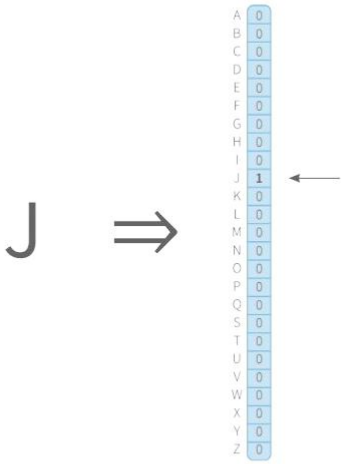

Figure 2 – Representation of the letter **“J”** in one-hot encoding

By treating each character as a distinct dimension, the problem of unwanted relationships between them is non-existent. In fact, any relationship between the characters is eliminated, since each dimension is orthogonal to all the others, consequently there is complete linear independence between them.

3.2.1 Word level

Using one-hot encoding, it is possible to have a representation of texts without introducing unwanted relationships between characters. In this encoding, a translation model would be input character by character. Similarly, the translation generated as output would also be returned character by character.

This introduces an additional difficulty to the translation task: not only must the model be able to interpret text and translate it, but it must also associate character sets with words. The model must be able to understand that the sequence of letters “W”, “O”, “R”, “L”, “D” forms the word “WORLD”, and that this word, in turn, refers to the Portuguese word “MUNDO”, for example.

Such an additional task of interpreting strings as meaningful words becomes an extra difficulty due to the encoding chosen in the previous section. Fortunately, this additional interpretation step is easily eliminated by changing the meaning of each dimension of the embedding vector.

It is important to emphasize that when human beings read texts, they are assimilating sequences of words, not of characters, and that such words represent ideas (WOOLLEY; WOOLLEY, 2011). These ideas and their relationships is what effectively makes up the content of a text. In this way, if a text is passed to the word-for-word translation model, it will have an easier time in its main task.

It is possible to use the same strategy as one-hot encoding to encode entire words. Each dimension of the one-hot vector will represent a word. Therefore, the dimension of this vector grows to the size of the vocabulary used in the *corpus*. This can cause vectors to have dimensions on the order of tens of thousands. Hence, it is necessary to be careful while working with vectors of this magnitude as it can make the algorithms slower, and it is essential to have more memory to store them [3].

3.3 Word embedding

Initially, using the one-hot representation of words appears to be a suitable method to represent a text since it resolves all the issues with previous methods. However, this encoding also has its own unique characteristics. Since each word is represented by a dimension that is perpendicular to all other dimensions, there is no association between two words in the vocabulary.

[3] https://www.defined.ai/blog/the-challenge-of-building-corpus-for-nlp-libraries/

Although this accurately represents the relationship between “cat” and “cement”, this encoding also implies that “cat” and “animal” have no connection, or that “cat” has the same level of relation to “animal” as it does to “cement”, as demonstrated in Figure 3. There are even more serious instances where words such as “cat” and “cats” have no correlation.

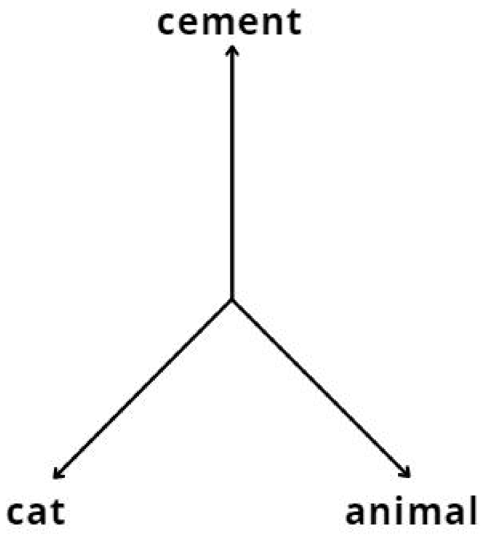

Figure 3 – Representation of “cat”, “animal” and “cement” in a vector space

One major advantage of using word embeddings instead of just strings is that word embeddings can capture semantic and syntactic relationships between words. In other words, similar words will have similar vector representations and be closer together in the embedding space.

This means that the embedding captures some of the meaning of the word beyond just its spelling. For example, in a well-trained word embedding, the vectors for “cat” and “dog” will be much closer together than the vectors for “cat” and “computer”. This can be very useful for many NLP tasks, such as sentiment analysis or language translation, as it allows models to better understand the meaning of words and their relationships to each other.

Another advantage of using word embeddings is that they can greatly reduce the dimensionality of the data. Strings are high-dimensional objects, where each character is represented by a unique code point or byte. In contrast, word embeddings typically have a much lower dimensionality, such as 100, 200, or 300 dimensions. This makes it easier to work with the data and improves the efficiency of models that use the embeddings. Additionally, the lower dimensionality of the embeddings reduces the risk of over fitting, as there are fewer parameters to learn [4].

[4] http://veredshwartz.blogspot.com/2016/01/representing-words.html

When a person analyzes a language, they can identify the dependencies and relationships between words' meanings. However, if these semantic relationships are not considered when creating a computational model, it can negatively impact the model's performance. In such a scenario, the model would need to create these relationships internally.

Natural Language Processing (NLP) experts have developed techniques for generating compact vector representations of words (PILEHVAR; CAMACHO-COLLADOS, 2020). In this method, each dimension of the vector denotes not the word itself, but characteristics that are shared among words. It becomes more evident when considering that the vector space has directions representing various linguistic and semantic concepts, including grammatical ones such as gender, number, and verb tense, and semantic concepts such as relationships between capitals and countries, as well as others (MIKOLOV; YIH; ZWEIG, 2013).

One of the main advantages of vector spaces and semantic relationships is the ability to perform analogical reasoning. This means that we can use vector arithmetic to solve word analogies such as "man is to woman as king is to ...", as illustrated in Figure 4. By subtracting the vector of "man" from the vector of "king" and adding the resulting vector to the vector of "woman", we can obtain a vector that is very close to the vector of "queen". This ability to reason analogically is a powerful tool in NLP and allows for more complex language understanding in ML models.

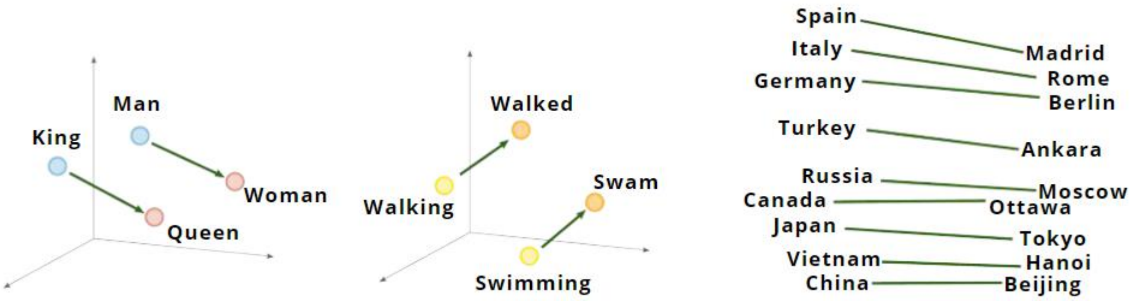

Figure 4 – Different directions in vector space with semantic matches. Adapted from (NSS, 2017)

After demonstrating the semantic and grammatical potential of word embeddings, the question arises: how can we create such vector spaces? There are several methods to generate these word vector spaces, but two of the most widely used are Word2Vec (MIKOLOV et al., 2013) and FastText (BOJANOWSKI et al., 2017).

Word2Vec learns to predict the context words for a given input word, thus capturing the word's meaning and relationships to other words. FastText, on the other hand, extends the Word2Vec model by also considering sub-word information, enabling it to generate embeddings for rare and out-of-vocabulary words by decomposing them into smaller sub-words.

3.3.1 Word2Vec

The methods are generally based on the so-called "distributed hypothesis" presented at (HARRIS, 1954). This hypothesis attests that words that appear in similar contexts tend to have similar meanings. Another possible interpretation is to say that words that are preceded and followed by other words in common tend to have similar meanings.

The first method to become popular for generating word embeddings was Word2Vec (MIKOLOV et al., 2013), introduced in 2013. The algorithm has two versions, which vary according to the output expected from each of them.

3.3.1.1 Continuous Bag-of-Words

The first version of Word2Vec was known as "Continuous Bag-of-Words", or CBOW, and has the following proposal: predict a central word given its context, that is, predict a central word preceded and followed by n words. For example, in the sentence "the cat jumped off the chair", if we consider "jumped" as the central word, the goal is to get "jumped" with n variables. If $n = 1$, we have "cat" and "off" in this case, and we get the result illustrated in Figure 5.

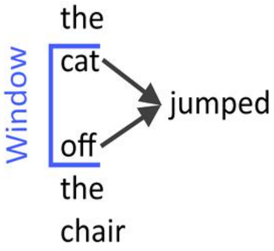

Figure 5 – CBOW representation of the sentence "the cat jumped off the chair". Taken from (SOLUTIONS, 2016)

Therefore, the objective is to maximize the probability that the word "jumped" will be returned by the model, given that the words "cat" and "off" have appeared. For a general understanding of the method, Figure 6 presents a more abstract illustration of this objective.

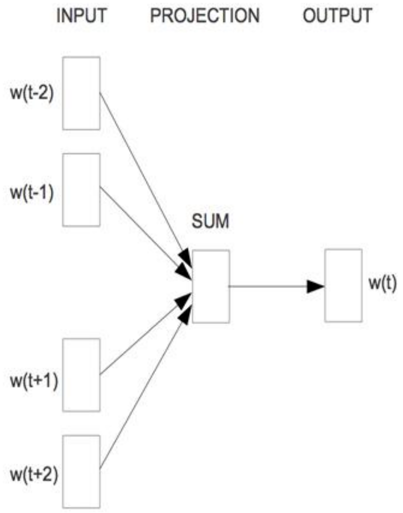

Figure 6 – CBOW generic representation. Taken from (MIKOLOV et al., 2013)

Let w_t be the central word of the text at position t, M the last possible position, and w_i be the word found at position i. We define P$(w_i|w_j)$ as the probability of the word w_i being returned, given that w_j appeared. The objective function that we want to maximize is given by:

$$\frac{1}{M}\sum_{t=1}^{M}\log P(w_t|w_{t-n},...,w_{t_1},w_{t+1},...,w_{t+n}). \tag{3.1}$$

Hence, we want to maximize the probability that, given the context words of the core word, we will generate the core word itself.

3.3.1.2 Skip-Gram

The second version, entitled Skip-Gram, tries to predict the context itself, given the central word. Going back to the previous example, the objective is to predict the words "cat" and "off", given the word "jumped" and $n = 1$, as portrayed by Figure 7.

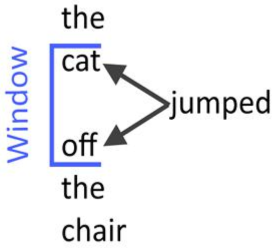

Figure 7 – Skip-Gram representation of the sentence "the cat jumped off the chair". Taken from (SOLUTIONS, 2016)

We present below Figure 8 as an illustration of the generic algorithm for Skip-Gram.

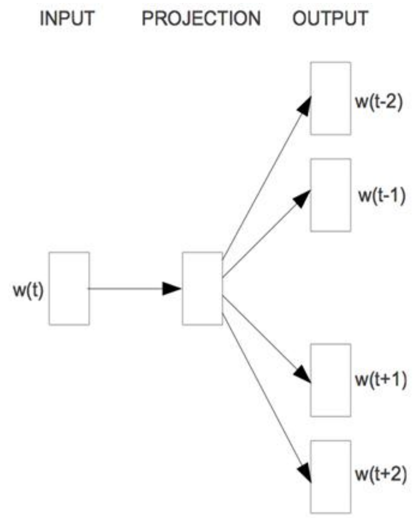

Figure 8 – Skip-Gram generic representation. Taken from (MIKOLOV et al., 2013)

Let n be the number of words that precede and follow them, used as context. The objective function we want to maximize is:

$$\frac{1}{M}\sum_{t=1}^{M}\sum_{-n\leq j\leq n;j\neq 0}\log P(w_{t+j}|w_t). \tag{3.2}$$

Therefore, the objective is to maximize the probability that, given a certain central word t, generates the context words $w_{t-n}, ..., w_{t_1}, w_{t+1}, ..., w_{t+n}$.

For both versions, probabilities can be calculated using the softmax function (GOLD; RANGARAJAN et al., 1996). Let W be the set of all words present in the text, and $\vec{w_o}^T\vec{w_i}$ the inner product between the vectors $\vec{w_o}$ and $\vec{w_i}$. We can consider the inner product as a certain degree of similarity between the two vectors. In the case of Skip-Gram, the probability of a word w_o being generated, given that the word w_i appeared is:

$$P(w_o|w_i) = \frac{e^{\vec{w_o}^T\vec{w_i}}}{\sum_{w_k\in\mathbf{W}} e^{\vec{w_k}^T\vec{w_i}}}. \tag{3.3}$$

If we used the gradient descent method (RUDER, 2016), commonly used in NLP, we would need to calculate the gradient of function 3.3. However, as written by the authors in (MIKOLOV et al., 2013), calculating the exact gradient is a computationally expensive operation. For this reason, the softmax function is often recommended for the translation task.

3.3.2 FastText

Finally, we present FastText, another existing method for generating embeddings, introduced in (BOJANOWSKI et al., 2017). According to the authors, their proposal is to be an extension of the Skip-Gram presented by (MIKOLOV et al., 2013), which takes information from word fragments into account.

In FastText, the vector of a word is represented as the sum of the representations of its fragments, for example, if we consider the size of each fragment equal to 4, the parts of the word "world" will be: "<wor", "worl", "orld" and "rld>". Therefore, the vector that represents "world" will be the combination of the vectors of each of these fragments. In order to differentiate words from fragments, it is necessary to add the markers "<" and ">" both at the beginning and at the end of the word.

The algorithm behind FastText is essentially the Skip-Gram presented in the previous section, with slight modifications. Let w_c be the context word, w_t be the central word and w_k each of the words present in W. We define F_t as the set of all fragments of the central word. In FastText, we calculate the inner product between the context word vector and each of the central word fragments as follows:

$$s(w_t, w_c) = \sum_{f \in F_t} \vec{w_f}^T \vec{w_c} \tag{3.4}$$

In this way, the probability of a word belonging to the context being generated, given the central word, is defined by:

$$P(w_c|w_t) = \frac{e^{s(w_t,w_c)}}{\sum_{w_k \in W} e^{s(w_t,w_k)}} \tag{3.5}$$

According to the authors, the model shares the representation of fragments between words, allowing the generation of high quality embeddings for rare words. A great advantage of this method is the possibility of obtaining good representations for words that were not present during the training of the algorithm, since this technique uses the vectors of the fragments of each word to generate the representations, it is likely that the word fragments of outside the vocabulary were present in the training. Hence, by combining these fragments that appeared previously in the training, FastText is able to provide good vectors for words not seen before.

3.3.3 Limitations

It is important to acknowledge that word embeddings also have certain limitations. A common example, presented by Figure 9, demonstrates that by adding the vector representing "king" to the vector representing "woman" and subtracting the vector representing "man", the resulting vector represents "queen" (MIKOLOV et al., 2013).

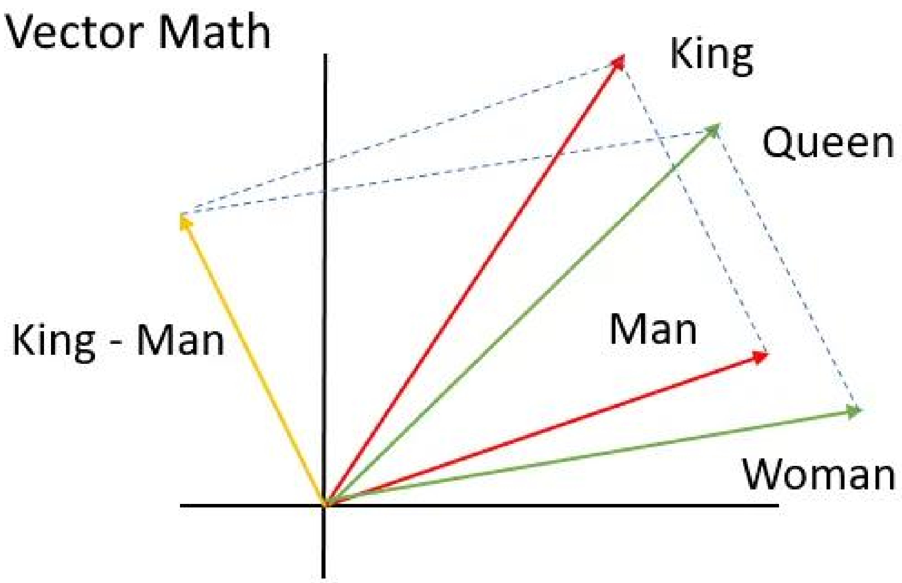

Figure 9 – Word algebraic example. Taken from (CR, 2020)

However, this result is only perfect in theory. In practice, we are dealing with an approximation of the outcome. When performing “king - man + woman”, we obtain a vector that, in the space generated by the word embedding, is close to the vector representing the word “queen”. Hence, we can observe that by adding and subtracting words using word embeddings, we obtain a vector that captures the sense of the operation, and by examining which words are close to this vector, we can derive a meaningful response.

Another drawback is that it inherits the inaccuracies and biases present in the word embeddings used. Word embeddings are learned from large *corpora* of text data, and this data may contain biases or skewed representations of certain concepts or groups. For instance, if the training data predominantly associates the term “worker” with male gender references, the word embedding model may learn to encode a gender bias. Similarly, if the training data contains imbalances or stereotypes, these biases can be reflected in the resulting word embeddings (PETRESKI; HASHIM, 2023).

The Table 1 [5] showcases the resulting words obtained from the word embedding calculation when performing the operation “worker - man + woman”. It demonstrates the words that are most similar to the result of this operation, indicating the associations captured by the word embeddings.

[5] Taken from https://github.com/RafaelxFernandes/Stanford-CS224N-2019/blob/main/a1/exploring_word_vectors.ipynb

Worker - Woman + Man	**Worker - Man + Woman**
workers	employee
employee	workers
working	nurse
laborer	pregnant
unemployed	mother
job	employer
work	teacher
mechanic	child
worked	homemaker
factory	nurses

Table 1 – Comparing bias

The obtained word embedding calculation result reveals an interesting observation in the form of the words “nurse” and “pregnant” being listed alongside terms like “employee” and “mother” for women, while words such as “unemployed”, “mechanic”, and“factory” are associated with men. This juxtaposition raises concerns about the biases and limitations inherent in the word embeddings and the training data used to generate them.

It is important to note that these associations are not only oversimplifications but also perpetuate stereotypes by implying that being a woman is linked to certain professions or life experiences, while men are associated with others. These findings highlight the need for critical analysis and caution when interpreting word embedding results, as they may inadvertently reinforce societal biases and preconceptions.

These biases can propagate to the sentence representations we generate, potentially reinforcing or perpetuating societal biases and inequalities. For example, if we use our approach to calculate the representation of a sentence containing the words “hardworking” or “ambitious”, the resulting vector may be influenced by the biases present in the word embeddings. This can lead to unintended consequences when using these representations in downstream applications, such as automated hiring processes or sentiment analysis.

It is crucial to be aware of these limitations and biases when working with word embeddings and utilizing them for sentence representation. Researchers and practitioners should take steps to mitigate biases in the training data and carefully consider the implications of using word embeddings in sensitive applications.

However, despite these limitations, it is important to acknowledge the ongoing efforts in the field to address the challenges associated with word embeddings and sentence representation. Researchers and practitioners are actively working on developing techniques to mitigate biases in training data and improve the overall fairness and unbiased nature of representations (PAPAKYRIAKOPOULOS et al., 2020; GARG et al., 2018). Debiasing methods and the incorporation of more diverse training data sources are among the approaches being explored to ensure that word embeddings are used responsibly and ethically in sensitive applications of NLP (BOLUKBASI et al., 2016; ZHAO et al., 2017; GONEN; GOLDBERG, 2019).

While Word2Vec remains a widely adopted and well-established approach for word embeddings, similar versions have emerged, offering promising avenues for capturing the semantic representation of entire sentences or documents. Some of these techniques, such as Paragraph Vector (LE; MIKOLOV, 2014b) or Skip-Thought Vectors (KIROS et al., 2015), involve training neural network models on large *corpora* to learn representations for sentences, allowing for a deeper understanding of their meaning. By considering the context and interplay of words within a sentence, they have the potential to capture the nuanced and complex semantics that individual word embeddings may miss.

There is also Sentence2Vec (LE; MIKOLOV, 2014a), which techniques aim to learn representations for sentences, accurately capturing the intricacies of sentence meaning presents challenges due to the complexity of language and varying contextual factors. As researchers continue to search into the development and improvement of Sentence2Vec, they must address these challenges to ensure its effectiveness in accurately capturing the rich meaning embedded in sentences. Continued advancements in Sentence2Vec hold the potential to revolutionize natural language processing by providing more comprehensive and nuanced representations of text.

Although these techniques show promise in capturing sentence-level semantics, it is important to note that they are still relatively new approaches and have not yet reached the same level of maturity and widespread adoption as word embeddings. In our study, we decided to utilize FastText for several reasons. Firstly, FastText has been extensively studied and widely adopted in the NLP community. It has proven to be effective in capturing word-level semantics and producing high-quality word embeddings. Additionally, FastText benefits from a wealth of available pre-trained models and resources, which can be readily applied to various tasks. In this work we are using embeddings trained on Common Crawl[6] and Wikipedia[7] provided by Facebook .

Furthermore, by working at the word level rather than the sentence level, we can leverage the rich linguistic information contained in individual words and their relationships. While sentence-level representations have their merits, word-level embeddings offer more fine-grained control and flexibility in capturing the nuances of language.

3.4 Vector Space Alignment

The creation of vector spaces using word embeddings has revolutionized the field of NLP, enabling new applications such as MT, sentiment analysis, and text classification (MIKOLOV et al., 2013). One of the most powerful features of these vector spaces is

[6] https://commoncrawl.org/

[7] https://www.wikipedia.org/

their ability to capture the semantic and syntactic relationships between words (BARONI; DINU; KRUSZEWSKI, 2014). These relationships can be exploited to perform tasks such as finding synonyms, identifying related words, and even translating between languages.

In order to use these vector spaces for translation purposes, we need to align them across languages. One popular technique for this is using a solution of the Orthogonal Procrustes Problem (SCHÖNEMANN, 1966), which is a method used to find the best transformation that aligns two sets of points in Euclidean space. In the context of word embeddings, Orthogonal Procrustes Problem is used to find a linear transformation that aligns the vector spaces of two languages.

The alignment of vector spaces using the Orthogonal Procrustes Problem allows for a mapping between words in different languages, which can be used for MT. For example, by aligning the vector spaces of English and French, we can create a mapping between English words and their French counterparts. This mapping can then be used to translate text from English to French and vice versa (PENNINGTON; SOCHER; MANNING, 2014).

To align the word embeddings of different languages, a dictionary of equivalents between the languages is required. This dictionary provides the necessary information to create a mapping function that aligns the embeddings. In the supervised approach to the Orthogonal Procrustes Problem, a pre-existing dictionary of language equivalents is utilized during the training phase (DEV; HASSAN; PHILLIPS, 2020). This dictionary can be prepared by either humans or machines.

On the other hand, the non-supervised approach can align the embeddings without prior knowledge of the language relationships. In this approach, the alignment process involves iteratively creating and enhancing a dictionary and mapping function (CONNEAU et al., 2018). In our study, as we had access to a large dataset with equivalent sentences in multiple languages, we opted for the supervised approach to align the word embeddings.

3.4.1 Orthogonal Procrustes Problem

The Orthogonal Procrustes Problem, presented in (SCHÖNEMANN, 1966), is about finding an orthogonal matrix [8] that maps a given set of points closest to another given set of points; the one-to-one correspondence of points between the two sets must be known *a priori*. The nomenclature refers to a character in Greek mythology, Procrustes, a bandit who stretched or cut off the limbs of his victims to fit his iron bed [9].

An orthogonal matrix is a square matrix R if and only if its transpose is the same as its inverse (SINAP; ASSCHE, 1996). Therefore, if we compute the dot product of R by its transpose the result will be the identity matrix, a characteristic that can be written

[8] Rowland, Todd and Weisstein, Eric W. “Orthogonal Matrix.” From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/OrthogonalMatrix.html

[9] https://www.britannica.com/topic/Procrustes

as $R^tR = I$. An orthogonal transformation R is a transformation that preserves the geometry (angles and distances).

Strictly speaking, if we consider two matrices A and B, as portrayed by Figure 10, the problem is to find an orthogonal transformation R that minimizes the difference between the two matrices. This can be written in an algebraic way as:

$$argmin_R \|RA - B\|^2, \text{ where } R^TR = I \tag{3.6}$$

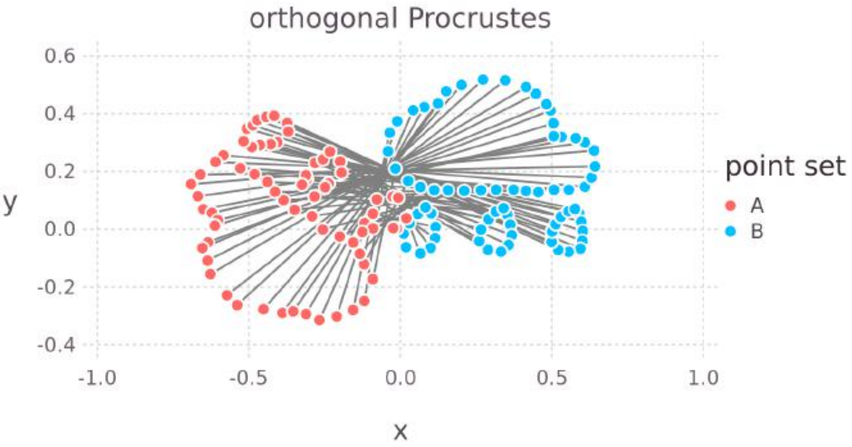

Figure 10 – Orthogonal Procrustes Problem. Taken from (SIMON, 2018)

Given that R is an orthogonal transformation, it does not change the length of the vector in the matrix it has been applied to. Moreover, the angle between the vectors does not change either. This is an important property of R since we want that the relations shown at the Word Embedding subsection to remain the same after the transformation.

3.4.1.1 Solution

In order to solve this problem, the appropriate approach is to use a function named "trace" [10], which is the sum of the diagonal values of a matrix. It can be denoted by tr and we will use its following properties to find a solution for the Orthogonal Procrustes Problem:

- $\|A\|^2 = tr(AA^T)$
- $tr(A^T) = tr(A)$
- $tr(AB) = tr(BA)$
- $tr(ABC) = tr(CAB) = tr(BCA)$

[10] Taboga, Marco (2021). "Trace of a matrix", Lectures on matrix algebra. https://www.statlect.com/matrix-algebra/trace-of-a-matrix.

- $tr(A+B) = tr(A) + tr(B)$

Using trace properties on equation 3.6:

$$\begin{aligned} argmin_R \|RA - B\|^2 &= argmin_R\ tr((RA-B)(RA-B)^T) \\ &= argmin_R\ tr((RA-B)((RA)^T - B^T)) \\ &= argmin_R\ tr((RA)(RA)^T - RAB^T - B(RA)^T + BB^T) \\ &= argmin_R\ tr((RA)(RA)^T) + tr(-RAB^T) + tr(-B(RA)^T) + tr(BB^T) \\ &= argmin_R\ tr(AA^T R^T R) + tr(-RAB^T) + tr(-((RA)B^T)^T) + tr(BB^T) \\ &= argmin_R\ tr(AA^T) + 2tr(-RAB^T) + tr(BB^T). \end{aligned} \tag{3.7}$$

Since we need R to minimize the result of the equation 3.6, a necessary transformation is to remove the negative sign, so we go from a minimization problem to a maximization one. Therefore, making these adjustments on equation 3.7:

$$\begin{aligned} argmin_R \|RA - B\|^2 &= argmin_R\ tr(AA^T) + 2tr(-RAB^T) + tr(BB^T) \\ &= argmin_R\ tr(-RAB^T) \\ &= argmax_R\ tr(RAB^T). \end{aligned} \tag{3.8}$$

This is an equivalent optimization problem. R is required to maximize $tr(RAB^T)$. Using equation 3.8, we are able to compute the Singular Value Decomposition (KALMAN, 1996), or SVD, of AB^T. This decomposition returns three matrices: U, V and Σ.

Σ is a diagonal matrix. A diagonal matrix is a matrix that is both upper triangular and lower triangular. i.e., all the elements above and below the principal diagonal are zeros and hence the name “diagonal matrix” (PARTER; YOUNGS, 1962).

U and V are both orthogonal matrices, meaning their norms are equal to 1. Norm is a function that returns the size of any vector, and is calculated by taking the square root of the sum of each component of the vector [11].

$$\begin{aligned} \text{with } AB^T \approx SVD(AB^T) &= U\Sigma V^T \\ argmax_R\ tr(RAB^T) &= argmax_R\ tr(RU\Sigma V^T) \\ &= argmax_R\ tr(V^T RU\Sigma) \\ &= argmax_R\ tr(Z\Sigma). \end{aligned} \tag{3.9}$$

With the aim of facilitating the demonstration, we renamed the $V^T RU$ matrix as Z. Since trace is an operation that depends on the diagonal and Σ is a diagonal matrix, we can rewrite equation 3.9 as:

[11] https://www.ime.unicamp.br/~marcia/AlgebraLinear/norma.html

$$argmax_R\, tr(Z\Sigma) = argmax_R\, tr([Z_{11}\Sigma_{11} \cdots Z_{nn}\Sigma_{nn}]). \tag{3.10}$$

We need to maximize $Z_{ii}\Sigma_{ii}$ for all values of i, without modifying Σ_{ii} value. Because of this reason, we will consider the Z matrix. As Z is a product of three orthogonal matrices, Z is also orthogonal, therefore its norm is equal to 1 as well. Hence, in order to select the highest values of the diagonal and to maintain the orthogonality properties, Z needs to be the identity matrix (GOETHALS; SEIDEL, 1967).

$$\begin{aligned} I &= Z \\ I &= V^T R U \\ R &= V U^T. \end{aligned} \tag{3.11}$$

$R = VU^T$ is the translator matrix we are looking for, with U and V being the result of the SVD of AB^T. Applying R on A results in a vector which approximately represents the vector space B, as illustrated by Figure 11. In case we want to translate from B to A, the transformation is as simple as R^T since V and U are orthogonal.

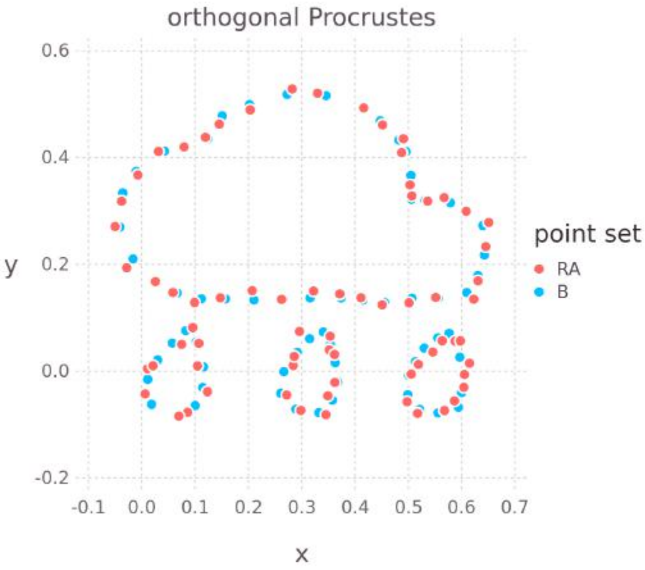

Figure 11 – Orthogonal Procrustes Problem solution. Taken from (SIMON, 2018)

It is important to note that A and B need to have the same number of rows and, for the transformation to be meaningful, the rows need to be ordered in a way that a given row with an index i in the matrix A is related with another row with the same index i in the matrix B. As a consequence of this fact, from our 16521 sentences per language, only 10941 are usable, which represents 66.22% of the total.

In order to evaluate our results, we apply three metrics commonly used in automatic translation tasks: cosine similarity (XIA; ZHANG; LI, 2015), Euclidean distance (KRISLOCK; WOLKOWICZ, 2012), and BLEU (PAPINENI et al., 2002). In the following sections we explain each one of them.

3.5.1 Cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It has been widely used in various applications, including text classification, information retrieval, and recommendation systems. One of the advantages of cosine similarity is that it is invariant to the length of the vectors, which makes it useful for comparing documents of different lengths.

Implementation of this metric can be applied to any two texts (sentence, paragraph, or whole document). Mathematically, cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional space, and the closer the sentences are by angle, the higher is the cosine similarity ($\cos\theta$), which is calculated by

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{3.12}$$

where $\vec{a}\cdot\vec{b} = \sum_{i=1}^{n} a_i b_i$ is the dot product of the two vectors.

3.5.2 Euclidean distance

Euclidean distance is another widely used distance measure that calculates the distance between two points in n-dimensional space, which is equivalent to the length of the straight line connecting those two points. It has been used in various applications, including clustering, anomaly detection, and outlier detection. However, one of the limitations of Euclidean distance is that it can be sensitive to outliers, which can affect the accuracy of the results.

Let us assume that $(x_1,\, y_1)$ and $(x_2,\, y_2)$ are two points in a two-dimensional plane. The Euclidean distance formula states:

$$d_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.13}$$

in which $(x_1,\, y_1)$ are the coordinates of one point, $(x_2,\, y_2)$ of the other point, and d_e is the distance between these two points. We use this formula when we are dealing with 2 dimensions. We can generalize this for an N-dimensional space as:

$$D_e = \sqrt{\sum_{i=1}^{N}(x_i - y_i)} \tag{3.14}$$

where N is the number of dimensions, and x_i and y_i are data points.

3.5.3 BLEU Score

Translation quality assessment plays a crucial role in the field of machine translation. To objectively measure the performance of machine-generated translations, various evaluation metrics have been developed. One prominent metric widely used in the research community is the BLEU (Bilingual Evaluation Understudy) score (PAPINENI et al., 2002).

BLEU is a precision-based metric that compares the machine-generated translation against one or more reference translations. It measures the similarity between the candidate translation and the references by computing the precision of n-gram matches. The BLEU score (BLEU) is computed as the geometric mean of the individual n-gram precision, with a brevity penalty (BP) applied to account for differences in translation length:

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

where N is the maximum order of the n-grams, w_n is the weight assigned to the n-gram precision, and p_n is the precision of n-grams. The brevity penalty BP is used to penalize translations that are shorter than the references.

In the context of BLEU, n-grams are contiguous sequences of words used to assess the quality of machine-generated translations. The precision of n-grams measures the overlap between the candidate translation and the reference translations. For example, if we consider a 2-gram precision, it compares pairs of consecutive words between the candidate and reference translations. By considering higher-order n-grams (e.g., 3-grams, 4-grams), BLEU captures not only individual word choices but also the coherence and fluency of longer sequences. This approach allows BLEU to evaluate translations at different levels of granularity, capturing both local and global aspects of translation quality.

The underlying principle of BLEU is to assess the quality of the translation output based on the degree of overlap between the machine-generated translation and the reference translations. By evaluating the precision of n-gram matches, BLEU captures the fidelity of the translation at the lexical level, giving higher scores for translations that contain similar n-grams as the references.

One of the strengths of BLEU is its simplicity and efficiency in providing a quantitative measure of translation quality. The score ranges from 0 to 1, with higher scores indicating better translation quality. However, it is important to note that BLEU has certain limitations. For instance, it does not account for semantic accuracy, syntactic structure, or overall fluency, as it primarily focuses on n-gram precision.

Despite its limitations, BLEU remains a widely used and accepted metric in the machine translation community. It serves as a benchmark for comparing different translation systems, fine-tuning models, and evaluating the progress of machine translation research. Researchers often report BLEU scores to provide objective measurements of translation quality and to facilitate meaningful comparisons with previous work.

4 Methodology

The Methodology chapter of this study encompasses various components related to data collection, study cases and the step-by-step approach employed. It aims to present an overview of the research process and the specific procedures followed to achieve the study's objectives, and to establish transparency to serve as the basis for the subsequent analysis and interpretation of the research findings.

4.1 Dataset Description

In order to facilitate translation between multiple languages, a large dataset with diverse language pairs is essential. For this purpose, we utilized the MASSIVE dataset developed by Amazon [12]. MASSIVE comprises a vast collection of 1 million realistic, parallel, and labeled virtual assistant utterances, encompassing a broad range of linguistic aspects.

The MASSIVE dataset provides extensive coverage across multiple dimensions. It was created to translate the English-only SLURP dataset (BASTIANELLI et al., 2020) into 50 languages from diverse language groups (FITZGERALD et al., 2022). It encompasses 51 languages, representing a wide range of linguistic diversity from various language families such as Romance, Germanic, Slavic, and more.

Furthermore, the dataset covers 18 domains, which refer to distinct subject areas or industries. These domains include technology, travel, healthcare, finance, sports, and others.

The dataset also includes 60 intents, which represent the underlying purposes or goals of user interactions. These intents can vary from booking reservations, seeking information, making recommendations, to navigation-related queries.

Additionally, the dataset consists of 55 slots, which represent specific pieces of information within user utterances. These slots can refer to dates, times, locations, names, and other relevant details. In summary, the MASSIVE dataset's remarkable coverage of languages, domains, intents, and slots provides a foundation for evaluating and analyzing translation performance across diverse linguistic contexts, subject areas, user intents, and information types.

MASSIVE is a parallel dataset, meaning that every data entry is provided in all 51 languages included in the dataset (FITZGERALD et al., 2022). This unique characteristic of MASSIVE enables models to learn shared representations of utterances with the same intents, regardless of the language in which they are expressed. Such parallel data

[12] https://github.com/alexa/massive

facilitates cross-linguistic training on various Natural Language Understanding (NLU) tasks.

The availability of parallel data in MASSIVE also opens up opportunities for adaptation to other NLP tasks beyond NLU. For instance, the parallel nature of the dataset makes it suitable for machine translation, where the shared representations learned from the parallel utterances can be leveraged to improve translation quality across different languages (FITZGERALD et al., 2022).

Furthermore, the dataset can be utilized for multilingual paraphrasing, enabling the generation of diverse paraphrases in multiple languages. Additionally, the parallel structure of MASSIVE allows for new linguistic analyses of imperative morphologies, facilitating a deeper understanding of the grammatical and semantic properties of different languages (BASTIANELLI et al., 2020).

The MASSIVE dataset is structured as a collection of JSON [13] text files where each line represents a separate data entry, organized by locale following the ISO-639-1 convention (BYRUM, 1999). Each locale has its dedicated file, encompassing all dataset partitions.

It consists of several attributes that provide valuable information about each data entry. These attributes include the “id”, “locale”, “utt” and “worker_id”. The meanings and descriptions of these attributes can be found in Table 2.

Attribute	**Description**	**Example**
id	Maps to the original ID in the SLURP collection	12345
locale	Language and country code according to ISO-639-1	en-US
utt	Raw data entry text without annotations (or “utterance”)	“What’s the weather like today?”
worker_id	Obfuscated worker ID from Amazon Mechanical Turk	MT12345678

Table 2 – Attributes of the MASSIVE dataset

This research is centered around conducting in-depth translation tasks and conducting comparisons. The primary objective is to analyze and compare translations using the dataset’s data entries, available in the “utt” attribute. This attribute exclusively contains the raw sentences without any additional annotations or accompanying information.

4.2 Selection of Languages for the Study

In addition to the aforementioned details, it is worth highlighting the selection criteria for the specific languages used in this study. Three languages from the Romance language

[13] https://www.json.org/json-en.html

family and three from the Germanic language family were chosen for the study cases and experiments. This decision was influenced by our background as Brazilians writing in English. By selecting languages we were more familiar with, we aimed to enhance our understanding of the translation process and minimize potential language-related biases or challenges.

Language	**Language Family**
Portuguese	Romance
Spanish	Romance
Italian	Romance
English	Germanic
German	Germanic
Swedish	Germanic

Table 3 – Selected Languages for Study

Table 3 presents three languages — Portuguese, Spanish, and Italian — that belong to the same language family characterized by historical and linguistic roots in the Romance language. These languages have evolved independently over time and are widely spoken in various countries. They possess a rich cultural heritage and share common linguistic origins (PENNY; PENNY, 2002; POSNER et al., 1996; MAIDEN, 2014). Furthermore, these languages benefit from a wealth of linguistic resources and research materials that facilitate in-depth analysis and investigation (POSNER et al., 1996).

Portuguese, as the official language of Brazil, Portugal and other countries, possesses a vast corpus of written and spoken texts, including literature, newspapers, and online content, making it a valuable language for translation studies (SARDINHA; FERREIRA, 2014). Spanish, spoken in Spain and various Latin American countries, boasts a large number of native speakers and an extensive body of literature and linguistic data for analysis (PENNY; PENNY, 2002). Italian, the official language of Italy and one of the Romance languages, has a rich literary tradition and a wealth of linguistic resources that facilitate research (MAIDEN, 2014). The availability of ample linguistic resources and materials for Portuguese, Spanish, and Italian contributes to the suitability of these languages for conducting translation experiments and comparisons within the context of this study.

On the other hand, the three languages chosen from the Germanic language family are: English, German, and Swedish. They have developed independently over time and are widely spoken in different countries. English, in particular, has become a global lingua franca and has extensive resources and data available on the internet for analysis and research (CRYSTAL et al., 2003; BAKER, 2018). German and Swedish also benefit from a considerable amount of linguistic data and resources, allowing for study and experimentation (SCHMID, 2010; GANUZA; HEDMAN, 2015). The abundance of linguistic resources and materials for these languages enhances the practicality and credibility of conducting translation experiments and making comparisons within this particular set of languages.

In order to ensure a better exploration of the translation process, it was crucial to adopt a balanced approach in the selection of languages for the study cases. By including three languages from each language family, we were able to create a diverse range of translation permutations while maintaining a manageable scope for the experiments. This selection not only facilitated evaluations and comparisons but also optimized the available resources and minimized potential biases. With this balanced representation, we can now proceed to present the specific study cases, examining their objectives and methodologies in detail.

4.3 Study Cases

This section presents the selection and categorization of the cases under study. It outlines the criteria used to identify and classify the cases, ensuring the representation of diverse scenarios and variables relevant to the research objectives.

It explores two distinct cases that demonstrate the applicability of our translation approach. Each case represents a unique scenario that allows us to analyze and compare different translation techniques. These cases include “Direct Translation”, “Indirect Translation”, “Translation with Single Language Family Change” and “Translation with Double Language Family Change” as subsets of indirect translation. In the following subsections, we will present each case individually, examining the specific objectives and methodologies.

To facilitate comprehension, the examples in this section will be based on the sentence “Wake me up at five in the morning this week”.

4.3.1 Direct Translation

In this study case, our primary focus is to translate languages within the same language family. Our aim is to investigate and compare translations between languages within each respective family, examining the unique characteristics and challenges associated with these language groups.

Table 4 presents an example of a sentence from the Amazon MASSIVE dataset translated from Portuguese to Spanish. The first column contains a sample sentence extracted from the Portuguese language dataset, while the second column displays the actual translation in Spanish.

Similarly, Table 5 provides an example of a sentence translated from English to German. The first column presents a sentence from the English dataset, and the second column showcases its corresponding translation in German.

Sentence in Portuguese	Translation in Spanish
Acorda-me às cinco da manhã durante esta semana	Despiértame a las cinco de la mañana esta semana

Table 4 – Sentence Translation from Portuguese to Spanish

Sentence in English	Translation in German
Wake me up at five in the morning this week	Weck mich diese Woche um fünf Uhr morgens auf

Table 5 – Sentence Translation from English to German

4.3.2 Indirect Translation

Now we extend our study to include an intermediate language from the same family. This addition introduces an additional layer of complexity as we explore translation performance in a cascaded manner. By considering language combinations such as Portuguese-Italian-Spanish and English-Swedish-German, we investigate the impact of introducing an intermediate language that belongs to the same family on the overall translation quality.

For each language combination, we select one language as the source, another as the target, and the remaining language as the intermediary. We then perform cascaded translations, where the source language is first translated into the intermediary language and then further translated into the target language. This cascaded translation approach allows us to analyze the similarities and differences between direct translations and those involving an intermediate step.

Tables 6 and 7 showcase an illustrative example of sentence translations from the Amazon MASSIVE dataset. The tables consist of three columns, with each column representing the source sentence in its language.

Sentence in Portuguese	Passing by Italian	Translation in Spanish
Acorda-me às cinco da manhã durante esta semana	Svegliami alle cinque del mattino questa settimana	Despiértame a las cinco de la mañana esta semana

Table 6 – Sentence Translation from Portuguese to Spanish, with Italian as Intermediate Language

Sentence in English	Passing by Swedish	Translation in German
Wake me up at five in the morning this week	Väck mig klockan fem på morgonen den här veckan	Weck mich diese Woche um fünf Uhr morgens auf

Table 7 – Sentence Translation from English to German, with Swedish as Intermediate Language

Building upon the knowledge gained from the translation experiments within the same language family, we now shift our focus to exploring the dynamics of multilingual translation with single language family change.

4.3.2.1 Translation with Single Language Family Change

In this subsection, we explore the translation process involving a single change in language family. Specifically, we investigate translations from one language within a particular language family to another language within a different family, passing through an intermediate language.

Through the selection of source, target, and intermediate languages, we construct language combinations that span diverse linguistic origins. This allows us to analyze the interplay between languages with distinct linguistic characteristics, identify potential challenges arising from the linguistic differences, and explore the potential benefits offered by an intermediate language.

We conducted translations from Portuguese to German, and from English to Italian, each one with Spanish and the other with Swedish as the intermediate language. These translation combinations were selected as illustrative examples to provide a better understanding for the reader. It is important to note that these examples do not represent the entire scope of our translation experiments, but rather serve as representative cases to showcase the translation process and highlight the role of an intermediate language. The translations are presented in Tables 8, 9, 10 and 11.

Sentence in Portuguese	**Passing by Spanish**	**Translation in German**
Acorda-me às cinco da manhã durante esta semana	Despiértame a las cinco de la mañana esta semana	Weck mich diese Woche um fünf Uhr morgens auf

Table 8 – Sentence Translation from Portuguese to German, with Spanish as Intermediate Language

Sentence in Portuguese	**Passing by Swedish**	**Translation in German**
Acorda-me às cinco da manhã durante esta semana	Väck mig klockan fem på morgonen den här veckan	Weck mich diese Woche um fünf Uhr morgens auf

Table 9 – Sentence Translation from Portuguese to German, with Swedish as Intermediate Language

Sentence in English	Passing by Spanish	Translation in Italian
Wake me up at five in the morning this week	Despiértame a las cinco de la mañana esta semana	Svegliami alle cinque del mattino questa settimana

Table 10 – Sentence Translation from English to Italian, with Spanish as Intermediate Language

Sentence in English	Passing by Swedish	Translation in Italian
Wake me up at five in the morning this week	Väck mig klockan fem på morgonen den här veckan	Svegliami alle cinque del mattino questa settimana

Table 11 – Sentence Translation from English to Italian, with Swedish as Intermediate Language

4.3.2.2 Translation with Double Language Family Change

Expanding our exploration of the translation process, we now introduce translation with double language family change, offering a different perspective on multilingual translation dynamics. Building upon our previously outlined translation approach, known as the cascaded methodology, we employ a two-step process. The source language is first translated into the intermediary language before progressing to the translation into the target language.

Tables 12 and 13 present illustrative examples of sentence translations that involve multiple language families. These examples showcase the translation process from one language family to another and then back to the source language family.

Sentence in Portuguese	Passing by English	Translation in Spanish
Acorda-me às cinco da manhã durante esta semana	Wake me up at five in the morning this week	Despiértame a las cinco de la mañana esta semana

Table 12 – Sentence Translation from Portuguese to Spanish, with English as Intermediate Language

Sentence in English	Passing by Portuguese	Translation in German
Wake me up at five in the morning this week	Acorde-me às cinco da manhã esta semana	Weck mich diese Woche um fünf Uhr morgens auf

Table 13 – Sentence Translation from English to German, with Portuguese as Intermediate Language

4.4 Method

In order to streamline our development process and avoid redundancy, we opted to construct a dictionary that encompasses all the sentences from the MASSIVE dataset that we intend to utilize, along with their corresponding vector representations. This dictionary serves as a fundamental resource for our research, enabling efficient retrieval and storage of sentence vectors.

To achieve this goal of constructing a dictionary that encompasses all the sentences from the MASSIVE dataset along with their vector representations, we employ a method that leverages the power of word embeddings to transform the sentences into vector representations. By employing this approach, presented by Figure 12, we can capture the semantic and contextual information embedded within the sentences, enabling the analysis and exploration of translation techniques.

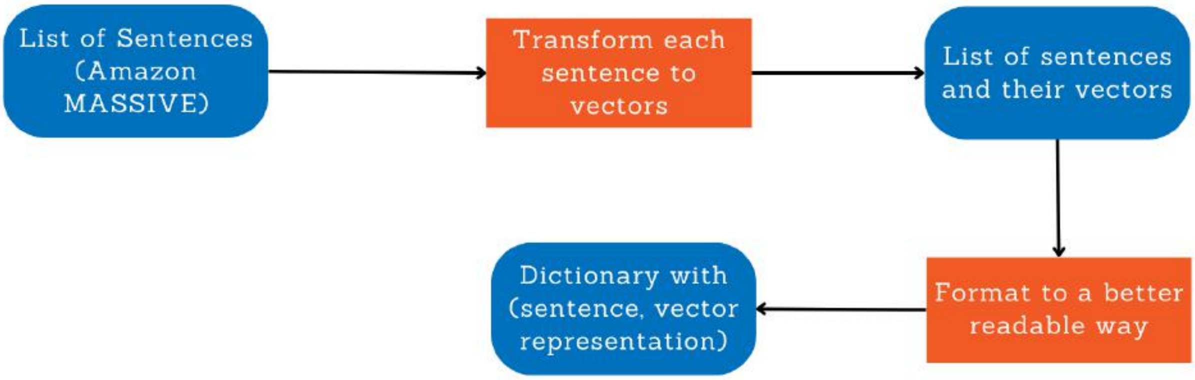
List of Sentences (Amazon MASSIVE)
Transform each sentence to vectors
List of sentences and their vectors
Format to a better readable way
Dictionary with (sentence, vector representation)

Figure 12 – Step by step creating the dictionary

4.4.1 Syntax Summation

One of the objectives of word embeddings is not only to create a numerical representation for words but also to build a vector space where vectors representing semantically similar words are located in close proximity to each other. By recognizing that sentences are composed of words and employing the fact that word embedding arithmetic yields vectors representing the meaning of the operation, we decided to sum the vectors of each word in a sentence to obtain its representation.

We introduce this approach to determine the syntactic similarity between sentences from different languages. Our method leverages the concept of vector space alignment and employs a combination of vector summation and Orthogonal Procrustes Problem to obtain a unified representation for sentence comparison.

We refer to this process as “Syntax Summation”. By summing the vectors of individual sentences, we create a composite representation that captures the collective syntactic information of the sentence pair. This summation operation enables us to create a shared space where sentences from different languages can be directly compared, facilitating cross-lingual similarity assessment.

Aggregating the word vectors allows us to capture the syntactic information of the sentence within a single vector representation. This approach offers a practical means of converting sentences into numerical representations. It enables the utilization of these representations in various downstream NLP tasks, including similarity comparison, clustering, and classification. Notably, this method, presented by Figure 13, eliminates the need for additional training or intricate algorithms, providing a straightforward and efficient solution for processing sentences in computational linguistics.

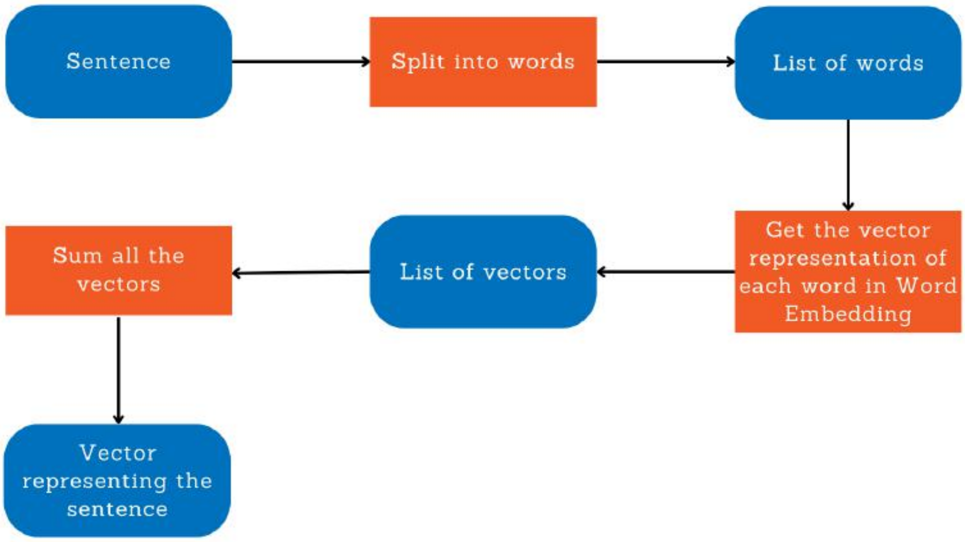
Sentence
Split into words
List of words
Get the vector representation of each word in Word Embedding
List of vectors
Sum all the vectors
Vector representing the sentence

Figure 13 – Step by step transforming sentences into vectors

4.4.1.1 Limitations

However, while our Syntax Summation provides a convenient and efficient method for representing sentences, it does have its limitations. One important limitation is the disregard for the contextual meaning of words. Words can often have multiple senses or meanings depending on the context in which they are used. For instance, consider the sentences “White, I like chocolate” and “I like white chocolate”. In the first sentence, “White” refers to a person, while in the second sentence it modifies the type of chocolate. Our approach fails to capture these subtle differences in meaning, as it treats both instances of “white” as identical.

Additionally, the disregard for word order in our representation calculation poses another limitation. By treating sentences as a collection of words without considering their sequential arrangement, we lose the syntactic and grammatical structure that contributes to the overall meaning of the sentence. For example, consider the sentences “The dog saw the boy” and “The boy saw the dog”. In both cases, the subject and object are reversed, resulting in different interpretations of the event. However, our approach would assign the same representation to both sentences, disregarding this crucial distinction.

4.4.2 Translation Process

One important step in our preparation phase is the calculation of translation matrices. These matrices play a crucial role in transforming the vectors representing sentences in the source language to their equivalent vectors in the target word embedding space. It is necessary to calculate a unique translation matrix for each language pair involved in our translation experiments.

The purpose of the translation matrix is to capture the linguistic mappings between two different languages. By aligning the vector spaces of the source language and the target language, the translation matrix allows us to project the semantic information encoded in the source language vectors into the target language vector space.

To calculate the translation matrix, we employ alignment techniques that leverage bilingual resources, such as parallel *corpora* or dictionaries. These resources provide pairs of sentences or words in the source language and their corresponding translations in the target language. By aligning the embeddings of these parallel sentences or words, we can estimate the transformation matrix that best aligns the two vector spaces.

It is important to note that the translation matrix is specific to each language pair, as different language combinations may exhibit unique linguistic characteristics and structural differences. Therefore, we calculate a distinct translation matrix for each language pair involved in our experiments.

By computing the translation matrices, we establish a bridge between different languages, enabling us to map the syntactic information encoded in the source language to the target language. This crucial step lays the foundation for our subsequent translation experiments across different languages. This process is presented by Figure 14.

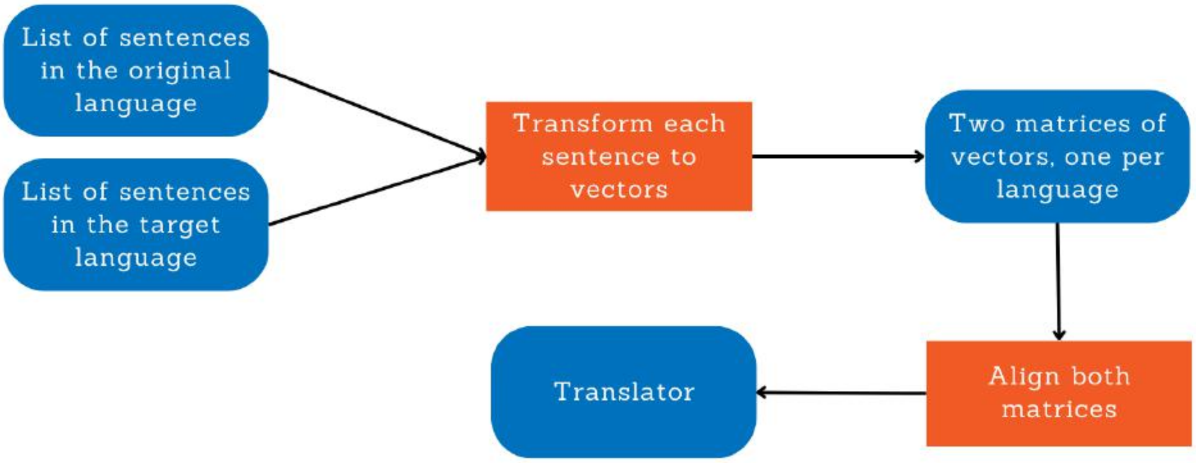
List of sentences in the original language
List of sentences in the target language
Transform each sentence to vectors
Two matrices of vectors, one per language
Align both matrices
Translator

Figure 14 – Step by step creation of translator

In the process of preparing our translation model, we undertake the computation of translation matrices, which play a crucial role in transforming sentence vectors from the source language to an equivalent representation in the target language. These matrices serve as a bridge between the two languages, facilitating the alignment and conversion of sentence-level semantics. To accomplish this, we construct two matrices: one where each row represents a sentence in the source language, and the other where each row represents a sentence in the target language. The sentences occupying corresponding positions in the matrices must be equivalent in meaning and content.

For instance, consider an example where the first row of the source language matrix represents the Portuguese phrase "Bom dia", which translates to "Good morning" in English. In the aligned translation matrix, the first row should reflect the corresponding translation, "Buenos días" in Spanish. This process ensures that the alignment between the two languages is consistent.

To derive these translation matrices, we allocate 30% of the total number of sentences available in our dataset, an arbitrary and common split for train data. This subset of sentences is aligned, ensuring that corresponding positions in both matrices contain the translated sentences.

After constructing these matrices, we utilize the pair of matrices to perform the steps outlined in the Orthogonal Procrustes Problem to obtain a transformation matrix that aligns the vector spaces. Prior to conducting our experiments, we calculate this matrix for all language pairs we intend to use and store it for future use in subsequent stages.

The Orthogonal Procrustes Problem addresses the challenge of finding an optimal transformation that aligns two sets of data. In our context, this problem aids in aligning the vector spaces of the source language and the target language. By applying the Procrustes algorithm to the constructed matrices, we derive a transformation matrix that captures the necessary adjustments and mappings required to align the semantic spaces.

Once we have obtained this alignment matrix, we store it for future reference and utilize it during the subsequent stages of our experiments. By precomputing and saving the alignment matrix for each language pair, we can streamline the translation process and reduce computational overhead in subsequent experiments.

One of the limitations of this approach is the requirement for a dataset of vectors representing phrases for which we know the translations in both languages. Fortunately, the MASSIVE dataset provided us with the necessary resources to automatically carry out this step. By leveraging this dataset, we were able to obtain a large collection of aligned sentence vectors in multiple languages, enabling us to proceed with the subsequent stages of our methodology.

On the positive side, the translation process in our approach is performed through matrix multiplication, a computationally efficient operation. This simplifies the translation process and allows for efficient processing by computers. By representing the sentence vectors as matrices and applying the alignment matrix obtained through the Orthogonal Procrustes Problem, we can obtain the translations through straightforward matrix operations. This computational efficiency is advantageous, as it enables us to handle large-scale translation tasks effectively and efficiently.

In order to perform the actual translation, we utilize the dictionary and translators that were computed in the previous steps. We retrieve the vector representation of a given sentence from the pre-calculated dictionary. Next, we select the appropriate translator that will map this vector from the source language's word embedding space to the word embedding space of the target language. Finally, we transform this vector representation back into a sentence in the target language. This process is presented by Figure 15.

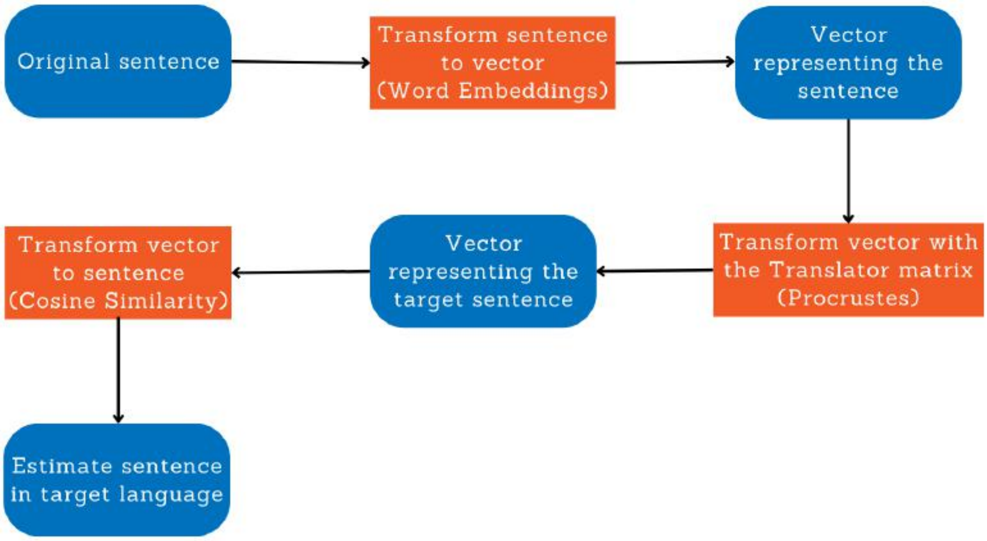
Original sentence
Transform sentence to vector (Word Embeddings)
Vector representing the sentence
Transform vector with the Translator matrix (Procrustes)
Vector representing the target sentence
Transform vector to sentence (Cosine Similarity)
Estimate sentence in target language

Figure 15 – Step by step of the translation process

With the translator and the vector representation of the sentence we want to translate at hand, we proceed to multiply the vector by the translator, resulting in a vector that represents the same meaning in the target space. To determine which sentence best represents this vector, we refer to the dictionary of the target language and find the sentence with the highest similarity. This similarity is calculated using cosine similarity. By employing cosine similarity, we can identify the most suitable translation candidate based on the similarity between vectors. This process ensures that the translated sentence captures the intended meaning as closely as possible, and is presented below by Figure 16.

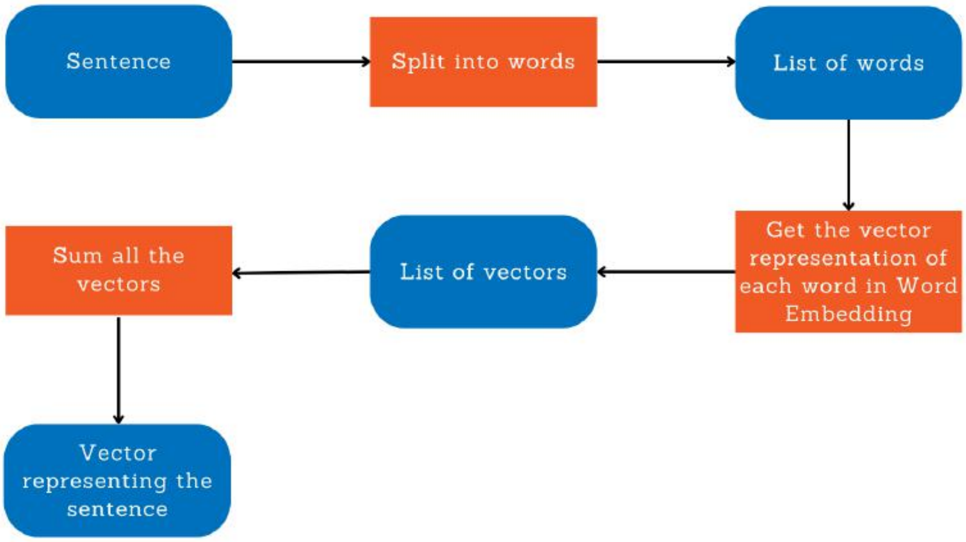
Sentence
Split into words
List of words
Get the vector representation of each word in Word Embedding
List of vectors
Sum all the vectors
Vector representing the sentence

Figure 16 – Step by step of the transformation of vectors into sentences

In conclusion, the methodological framework presented in this chapter lays the foundation for our exploration of multilingual translation, which will be used as shown in Figure 17. The forthcoming chapter of this work will be dedicated to presenting the results and findings of our experiments.

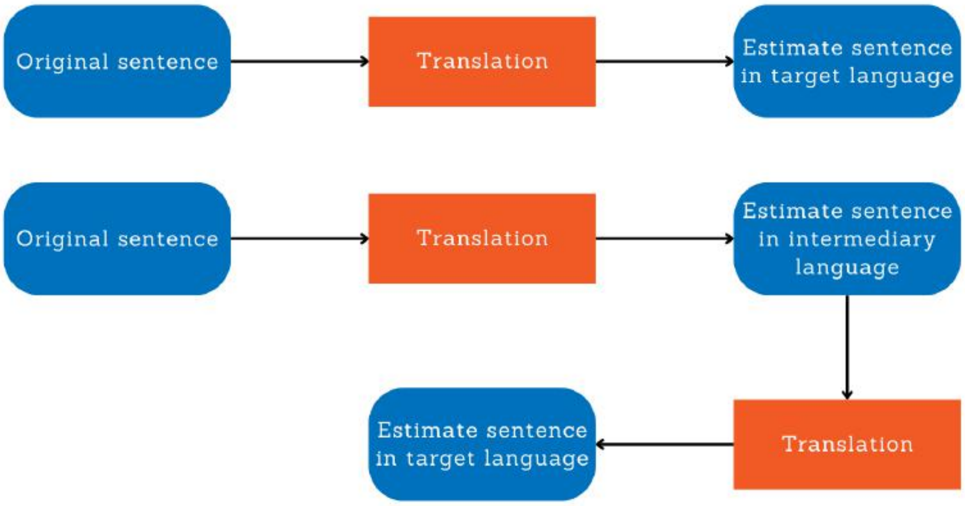
Original sentence
Translation
Estimate sentence in target language
Original sentence
Translation
Estimate sentence in intermediary language
Translation
Estimate sentence in target language

Figure 17 – Step by step of direct and indirect translations

5 Results

The specifications of the machine used for this study are detailed in Table 14. The entire process, which included loading the FastText files and running the experiments, was completed in approximately eight hours. It is important to note that the duration may vary based on factors such as the complexity of the translation tasks, the size of the dataset, and the computational resources available.

Component	**Specification**
Processor	Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz
RAM	16.0 GB (Usable: 15.8 GB)
Operating System	Windows 10 Pro version 22H2

Table 14 – Machine Specifications

After providing an overview of our methodology and the technical aspects involved, we can present the results obtained from our experiments. In order to evaluate the performance of our translation model, we started by conducting experiments for the following language pairs: Portuguese to Spanish, Portuguese to Italian, English to German, and English to Swedish.

The evaluation of our translation model's performance for different language pairs is presented in Tables 15, 16, 17, and 18. Each table follows a consistent structure, with the source language sentences in the first row, the model-generated translations in the second row, the target sentences in the respective target language in the third row, and the corresponding cosine similarity scores in the fourth row.

Portuguese sentence	remover uma lista (remove a list)
Spanish-generated sentence	eliminar una lista (delete a list)
Spanish sentence	borrar una lista (remove a list)
Cosine to spanish sentence	0.9128

Table 15 – Cosine similarity between sentences translated from Portuguese to Spanish

The results from the tables above suggest that our approach yields favorable outcomes. This finding aligns with our theoretical framework, indicating that our methodology holds promise in achieving translation results.

However, we can observe that the cosine similarity scores in Tables 16 and 17 are not equal to 1, despite the generated sentences being syntactically identical to the ones in the dataset. This discrepancy can be attributed to the distributional nature of word

Portuguese sentence	remover uma lista (remove a list)
Italian-generated sentence	rimuovi una lista (remove a list)
Italian sentence	rimuovi una lista (remove a list)
Cosine to italian sentence	0.9330

Table 16 – Cosine similarity between sentences translated from Portuguese to Italian

English sentence	remove a list
German-generated sentence	entferne eine liste (remove a list)
German sentence	entferne eine liste (remove a list)
Cosine to German sentence	0.8213

Table 17 – Cosine similarity between sentences translated from English to German

English sentence	remove a list
Swedish-generated sentence	skapa en lista till arbetet (make a list for work)
Swedish sentence	ta bort en lista (remove a list)
Cosine to Swedish sentence	0.8448

Table 18 – Cosine similarity between sentences translated from English to Swedish

embeddings and the way we are calculating the cosine similarity. While the generated sentence and the reference sentence may have the same words in the same order, their vector might not be precisely identical to the vector generated by our translation. Word embeddings capture contextual information and nuances, and even for perfectly translated words, slight variations in the vector representations can occur. As a result, the cosine similarity may not reach the maximum value of 1, despite the sentences being syntactically identical.

In Table 18, a noteworthy case came to our attention. Although the target sentence exhibited a notably high similarity score, the generated translation deviated from the intended context. Despite this divergence, the translated sentence maintained the overarching theme of lists; however, the specific action shifted from deletion to creation. This disparity can be attributed to the word embedding employed, which potentially clustered the terms "make" ("skapa") and "delete" ("ta bort") in close proximity, leading to a degree of confusion during the translation process.

Due to the inherent limitations of our approach and the data available, the translation results obtained in our experiments varied considerably in quality. Some translations were remarkably accurate, while others deviated significantly from the intended meaning. To present an overview, we will provide an example of a good, an average, and a bad result for each translation task.

In Table 19, we showcase a good translation result obtained in our experiments. These translations demonstrate a high level of accuracy and alignment with the intended meaning. On the other hand, Table 20 presents a bad translation outcome, where the generated sentences deviated the most from the desired translation.

Portuguese sentence	reproduz o podcast de hoje (play today's podcast from the mi)
Italian-generated sentence	riproduci il podcast di oggi (play today's podcast from the mi)
Italian sentence	riproduci il podcast di oggi (play today's podcast from the mi)
Cosine to Italian sentence	0.9464

Table 19 – A high cosine similarity example, obtained by translating from Portuguese to Italian

English sentence	play today's podcast from the mi
Italian-generated sentence	dai cinque stelle a questa canzone e salva il giudizio (rate this song five stars and save the rating)
Italian sentence	riproduci il podcast di oggi (play today's podcast from the mi)
Cosine to Italian sentence	0.4933

Table 20 – A low cosine similarity example, obtained by translating from English to Italian

Table 21 displays an average translation score, representing the typical performance of our methodology across various translation tasks. These results offer a balanced perspective, capturing the overall effectiveness of our approach in typical scenarios.

Portuguese sentence	põe o próximo podcast (play the next podcast)
Swedish-generated sentence	vad är nästa avsnitt spela det (what is the next episode in the podcast play it)
Swedish sentence	spela nästa podcast (play the next podcast)
Cosine to Swedish sentence	0.6886

Table 21 – An average cosine similarity example, obtained by translating from Portuguese to Swedish

Following our initial testing on single sentences, we made the decision to expand our evaluation to the entire test dataset. This broader evaluation allowed us to analyze the performance of our approach across a larger sample size, providing a better understanding of its effectiveness.

Our evaluation consisted of six distinct types of translations, and for each type, all permutations were calculated to explore the translation scenarios. The six types of translations considered in our evaluation were:

1. Direct within the Same Family: Source and target languages from the same language family (e.g., Portuguese to Italian)

2. Direct to Different Families: Source and target languages from different language families (e.g., Portuguese to English).

3. Using Intermediary Language within the Same Family: Source, intermediary, and target languages from the same language family (e.g., Portuguese to Spanish to Italian).

4. Using Intermediary Language with Source from Different Family: Source language from a different family, while intermediary and target languages are from the same family (e.g., English to Spanish to Italian).

5. Using Intermediary Language with Intermediary from Different Family: Source and target languages from same families, with the intermediary language from a different family (e.g., Portuguese to English to Spanish).

6. Using Intermediary Language with Target from Different Family: Source and intermediary languages from the same family, target language from a different family (e.g., Portuguese to Spanish to English).

As part of this evaluation, we utilized cosine similarity, Euclidean distance and BLEU as metrics to measure the effectiveness of our experiments. The choice to incorporate them in our evaluation stems from their different characteristics and common applications in translation tasks.

We considered cosine similarity because it is a suitable metric for evaluating word embeddings. Since word embeddings capture semantic relationships between words, cosine similarity is an appropriate measure to assess the semantic similarity between the machine-generated translations and the reference translations.

Euclidean distance was chosen as a metric because it provides a measure of dissimilarity or distance between vectors. In our evaluation, Euclidean distance offers an additional perspective on the dissimilarity between the generated translations and the reference translations.

BLEU is a widely used metric in machine translation tasks. By including BLEU in our evaluation, we can assess the quality of the machine-generated translations by evaluating their similarity to the reference translations, taking into account the precision of matching n-grams.

The results of our evaluation can be observed in Table 22. To accommodate the table, we simplified the column headings: “Cos. Sim.” represents the cosine similarity, and “Eucl. Dist” stands for Euclidean distance. Additionally, the values presented in the table are the means, and the standard deviation is indicated after each mean with a plus-minus symbol (±).

Type	Permutations	Cos. Sim.	Eucl. Dist.	BLEU
Direct within the same family	12	0.74 ± 0.02	4.51 ± 0.26	0.34 ± 0.06
Direct to different family	18	0.69 ± 0.02	5.02 ± 0.24	0.23 ± 0.06
Intermediary and all from same family	12	0.73 ± 0.02	4.60 ± 0.26	0.31 ± 0.05
Source from different family	36	0.69 ± 0.01	5.10 ± 0.26	0.21 ± 0.05
Intermediary from different family	36	0.72 ± 0.02	4.69 ± 0.24	0.30 ± 0.06
Target from different family	36	0.69 ± 0.02	5.10 ± 0.26	0.21 ± 0.06

Table 22 – Evaluation of Different Translations

When examining the table, higher values for cosine similarity and BLEU indicate better performance in terms of semantic similarity and translation accuracy, respectively. Conversely, for Euclidean distance, lower values are desirable, as they signify that the sentence vectors are closer in the embedding space, indicating better alignment and similarity between sentences.

The similar results observed in the cosine similarity scores between direct translation within the same family, using an intermediary with all languages from the same family, and using an intermediary with only the intermediary language from a different family raises intriguing questions about the translation process and the role of intermediary languages. One possible explanation for these findings is that when translating between languages of the same family and using an intermediary language, the vector space may act as a corrective mechanism. As the translation moves from one language to another and then back to the original source family, the vector space might help to rectify any errors or discrepancies introduced during the translation path.

Furthermore, the results suggest that, in certain cases, it may not be necessary to have all possible language pairs when using intermediary languages for translation. This finding has practical implications, as it implies that a more limited set of intermediary languages could still yield comparable translation performance, simplifying the translation process and reducing the need for a comprehensive collection of language pairs.

Additionally, the observations of a distinct group formed by direct translation to another family, using an intermediary language but source from another family, and using an intermediary language but target from another family with a mean cosine similarity of 0.69 indicate that when the source and target languages are from different families, the performance is likely to be worse, regardless of the intermediary language family. This highlights the importance of considering the language families involved in the translation process, as it can significantly impact the translation quality.

Although BLEU scores and cosine similarity are not directly comparable due to their different scales (BLEU ranges from 0 to 1, while cosine similarity ranges from -1 to 1), we can still analyze their relative patterns. The observations regarding BLEU scores follow a similar pattern to the cosine similarity, with higher scores aligning where expected. However, BLEU scores are generally lower compared to cosine similarity, and this discrepancy can be attributed to the difference in how BLEU is calculated. BLEU takes into account the order of words in a sentence, not its meaning, making it sensitive to word order variations. For instance, if the original sentence is "Please, play the next music", and the generated translation reads "Play the next music, please", the BLEU score may be low due to the word order difference, even though the cosine similarity remains high.

As mentioned in the previous chapters, our approach does not focus on maintaining the exact word order. Although BLEU is widely used in traditional translation tasks, our methodology emphasizes semantic representation over word order, making the cosine similarity a more relevant evaluation metric for our research.

Given this difference in evaluation metrics, our results cannot be directly compared with those of studies where BLEU is the main measure of translation quality. Instead, we emphasize the importance of using cosine similarity in our research to evaluate the semantic similarity between original and translated sentences, which aligns with our approach's objectives.

The project's source code, as well as instructions about how to run it, are available in a public repository on GitHub [14], and we encourage fellow developers and researchers to engage in further advancements and refinements in the field of NLP and multilingual communication.

[14] https://github.com/AlexSantoss/WordEmbedding-Translator/tree/3_Languages

6 Conclusion

The primary goal of this study was to explore the translation of short sentences using word embeddings and the orthogonal Procrustes problem. Through a comparative study, we investigated both direct and indirect translation approaches, where the latter involves the utilization of an intermediate language. In particular, we examined how translations performed when passing through an intermediary language, such as translating from Portuguese to English and then from English to Spanish, rather than a direct translation from Portuguese to Spanish.

Our findings revealed that indirect translations showed promising results in achieving meaningful translations. Interestingly, we observed comparable performance between direct translations within the same language family and indirect translations using an intermediary language from the same family. Moreover, we identified that word embeddings played a crucial role in capturing semantic relationships between languages, allowing for successful translation processes.

The implications of our research lie in the potential advancement of MT techniques. By incorporating an intermediate language, we can expand the applicability of translation systems, enabling efficient translation between languages with limited direct language pairs. This approach could be particularly valuable for languages with sparse resources, contributing to enhanced language accessibility and cross-lingual communication.

While our research presents promising results, we acknowledge certain limitations in our study. One limitation pertains to the choice of word embeddings and their potential biases or limitations in capturing the full semantic context. Additionally, the effectiveness of intermediate languages may vary depending on language families and the availability of bilingual data. These constraints provide opportunities for further investigation and refinement in future research.

The theoretical framework presented at this study has served as a guiding compass throughout our journey. Our findings align well with existing linguistic theories and concepts, supporting the notion that word embeddings can capture underlying semantic relationships between languages. This reaffirmation of our framework serves to validate the foundation of our study and its contribution to the broader body of knowledge in the field of MT.

Our methodology, utilizing word embeddings and the orthogonal Procrustes problem, proved effective in capturing the contextual information across languages. The use of embeddings allowed for a representation of language semantics, while the orthogonal Procrustes problem facilitated the alignment of different embeddings for cross-lingual translations. It is essential, however, to acknowledge that while we may not always achieve exact translations, our approach consistently captures the underlying context accurately.

We recognize that other approaches and algorithms may offer complementary insights and encourage researchers to explore alternative methodologies.

In addition to the examination of indirect translation with intermediate languages, our research introduced a novel approach called the “Syntactic Summation”. This approach leverages the power of word embeddings to capture not only the semantic but also the syntactic information embedded within sentences. The combination of intermediate language translation and the Syntactic Summation has proven to be a powerful methodology, enabling a better understanding of language relationships and enhancing the performance of our translation models.

Our initial hypothesis about the potential drawbacks of introducing an intermediary language has been confirmed by the results, revealing that such an approach can indeed have limitations in certain scenarios. However, our findings have also uncovered interesting nuances. Specifically, we observed that when the source and target languages are from the same language family, the impact of the intermediary language is not as significant. This highlights the importance of considering language relationships and the inherent similarities between related languages during the translation process.

Though our research might not revolutionize the industry, we believe it has contributed to the collective knowledge base in the field of MT. Moreover, it serves as a foundation for future investigations and inspires further exploration in the realm of language representations and translation techniques.

As we conclude this study journey, we are left with a sense of accomplishment and enthusiasm for the future of MT. Our work has brought to light the benefits of intermediate language translation, paving the way for more inclusive and efficient language communication across the globe. With this research as a stepping stone, we hope to further explore the realm of NLP and cross-lingual understanding.

6.1 Future works

In this section, we outline potential future directions and areas for improvement that could build upon the foundation laid by this research. While our study has provided valuable insights into intermediate language translation and its impact on MT, there are still several avenues worth exploring to enhance and expand upon our findings. The following paragraphs present key areas where future work can be pursued to enrich the capabilities and scope of our approach.

As a direction for future research, we intend to incorporate active learning techniques to further enhance the quality assessment of our translations. By involving experts in the field, we aim to gather valuable insights and refine our approach based on their feedback, ultimately contributing to the continuous improvement of our methodology.

In addition to these endeavors, we envision extending our testing to a broader array of languages and language families. This expansion will allow us to validate the conclusions drawn from our investigation into the use of intermediary languages and their impact on translation quality across diverse linguistic scenarios. Through this approach, we aim to deepen our understanding of the robustness of our findings and further substantiate the utility of our proposed translation method.

Continuing our trajectory, we aspire to further explore avenues for enhancing translation accuracy and efficiency. Our special attention will be directed towards meeting the requirements of marginalized communities and languages with limited resources. Emphasizing accessibility in MT will not only facilitate cross-cultural understanding but also contribute to fostering inclusivity and equality in the digital age.

Moreover, transforming our translation problem into a graph optimization problem opens up exciting possibilities for approaching the task in a novel and efficient manner. By representing languages as vertices and potential translation paths as edges, we can leverage graph algorithms to find the most optimal translation route between two languages. This approach enables us to explore various translation paths, taking into account factors such as language similarity, word embeddings, and intermediate languages.

Investigating the integration of advanced pre-trained models like BERT (DEVLIN et al., 2018) into our framework is an exciting avenue for future research. BERT's contextual embeddings offer a deeper understanding of language nuances and context, potentially allowing us to enhance translation accuracy significantly. The utilization of BERT, in conjunction with our existing methods, presents a promising research direction that holds the potential to elevate the quality of multilingual translations.

Furthermore, the incorporation of BERT into our methodology aligns with the broader trend in the NLP community towards utilizing transformer-based models for various language tasks. By adapting our approach to use the power of BERT, we aim to contribute to this evolving landscape and provide more robust translation solutions for diverse language pairs.

REFERENCES

ARNAUD, A. S. et al. Identifying cognate sets across dictionaries of related languages. 2017.

BAKER, M. **In other words: A coursebook on translation**. [S.l.]: Routledge, 2018.

BARENDREGT, H. P. et al. **Towards an intermediate language based on graph rewriting**. [S.l.]: Springer, 1987.

BARONI, M.; DINU, G.; KRUSZEWSKI, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. [S.l.: s.n.], 2014. p. 238–247.

BASTIANELLI, E. et al. Slurp: A spoken language understanding resource package. **arXiv preprint arXiv:2011.13205**, 2020.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. **Transactions of the association for computational linguistics**, MIT Press, v. 5, p. 135–146, 2017.

BOLUKBASI, T. et al. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. **Advances in neural information processing systems**, v. 29, 2016.

BYRUM, J. D. Iso 639-1 and iso 639-2: International standards for language codes. iso 15924: International standard for names of scripts. ERIC, 1999.

CERDA, P.; VAROQUAUX, G.; KÉGL, B. Similarity encoding for learning with dirty categorical variables. **Machine Learning**, Springer, v. 107, n. 8, p. 1477–1494, 2018.

CHEN, B. et al. Bilingual methods for adaptive training data selection for machine translation. In: **Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track**. [S.l.: s.n.], 2016. p. 93–106.

COHN, T.; LAPATA, M. Machine translation by triangulation: Making effective use of multi-parallel corpora. In: **Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics**. [S.l.: s.n.], 2007. p. 728–735.

CONNEAU, A. et al. **Word Translation Without Parallel Data**. 2018.

CR, A. **Word vectors**. 2020. https://medium.com/analytics-vidhya/word-embeddings-in-nlp-word2vec-glove-fasttext-24d4d4286a73 (Accessed on June 4, 2023).

CRYSTAL, D. et al. **English as a global language**. [S.l.]: Cambridge university press, 2003.

DAVIS, M.; COLLINS, L. Unicode. In: IEEE. **1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings**. [S.l.], 1990. p. 499–504.

DENKER, G.; MILLEN, J. K. Capsl intermediate language. In: CITESEER. **Proceedings of the Workshop on Formal Methods and Security Protocols—FMSP, Trento, Italy**. [S.l.], 1999.

DEV, S.; HASSAN, S.; PHILLIPS, J. M. **Closed Form Word Embedding Alignment**. 2020.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DONANDT, K.; CHIARCOS, C. Translation inference through multi-lingual word embedding similarity. In: **TIAD@ LDK**. [S.l.: s.n.], 2019. p. 42–53.

DOORSLAER, L. V. Journalism and translation. **Handbook of translation studies**, John Benjamins Amsterdam, v. 1, p. 180–184, 2010.

DREW, P. J.; MONSON, J. R. Artificial neural networks. **Surgery**, Elsevier, v. 127, n. 1, p. 3–11, 2000.

FITZGERALD, J. et al. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. **arXiv preprint arXiv:2204.08582**, 2022.

GANUZA, N.; HEDMAN, C. Struggles for legitimacy in mother tongue instruction in sweden. **Language and Education**, Taylor & Francis, v. 29, n. 2, p. 125–139, 2015.

GARG, N. et al. Word embeddings quantify 100 years of gender and ethnic stereotypes. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 115, n. 16, p. E3635–E3644, 2018.

GOETHALS, J.; SEIDEL, J. J. Orthogonal matrices with zero diagonal. **Canadian Journal of Mathematics**, Cambridge University Press, v. 19, p. 1001–1010, 1967.

GOLD, S.; RANGARAJAN, A. et al. Softmax to softassign: Neural network algorithms for combinatorial optimization. **Journal of Artificial Neural Networks**, Ablex Publishing Corp. Norwood, NJ, USA, v. 2, n. 4, p. 381–399, 1996.

GONEN, H.; GOLDBERG, Y. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. **arXiv preprint arXiv:1903.03862**, 2019.

GRAVE, E.; JOULIN, A.; BERTHET, Q. Unsupervised alignment of embeddings with wasserstein procrustes. In: PMLR. **The 22nd International Conference on Artificial Intelligence and Statistics**. [S.l.], 2019. p. 1880–1890.

GROUP, U. L. **5 Compelling Reasons to Use Machine Translation Tools**. 2018. https://www.unitedlanguagegroup.com/blog/compelling-reasons-to-use-machine-translation-tools (Accessed on August 15, 2023).

HARDWICK, J. C.; SIPELSTEIN, J. **Java as an intermediate language**. [S.l.], 1996.

HARRIS, Z. S. Distributional structure. **Word**, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

JANSSENS, M.; LAMBERT, J.; STEYAERT, C. Developing language strategies for international companies: The contribution of translation studies. **Journal of World Business**, Elsevier, v. 39, n. 4, p. 414–430, 2004.

KALMAN, D. A singularly valuable decomposition: the svd of a matrix. **The college mathematics journal**, Taylor & Francis, v. 27, n. 1, p. 2–23, 1996.

KENNY, D. Machine translation. In: **The Routledge handbook of translation and philosophy**. [S.l.]: Routledge, 2018. p. 428–445.

KIROS, R. et al. Skip-thought vectors. **CoRR**, abs/1506.06726, 2015. Disponível em: http://arxiv.org/abs/1506.06726.

KOEHN, P. **Statistical machine translation**. [S.l.]: Cambridge University Press, 2009.

KRISLOCK, N.; WOLKOWICZ, H. Euclidean distance matrices and applications. In: **Handbook on semidefinite, conic and polynomial optimization**. [S.l.]: Springer, 2012. p. 879–914.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: PMLR. **International conference on machine learning**. [S.l.], 2014. p. 1188–1196.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. **CoRR**, abs/1405.4053, 2014. Disponível em: http://arxiv.org/abs/1405.4053.

MAIDEN, M. **A linguistic history of Italian**. [S.l.]: Routledge, 2014.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MIKOLOV, T.; LE, Q. V.; SUTSKEVER, I. Exploiting similarities among languages for machine translation. **arXiv preprint arXiv:1309.4168**, 2013.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. **Advances in neural information processing systems**, v. 26, 2013.

MIKOLOV, T.; YIH, W.-t.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies**. [S.l.: s.n.], 2013. p. 746–751.

MUNDAY, J. Translation studies. **Handbook of translation studies**, John Benjamins Amsterdam, v. 1, p. 419–428, 2010.

NSS. **Semantic Vector Space**. 2017. https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/ (Accessed on May 12, 2022).

PAPAKYRIAKOPOULOS, O. et al. Bias in word embeddings. In: **Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency**. New York, NY, USA: Association for Computing Machinery, 2020. (FAT* ’20), p. 446–457. ISBN 9781450369367. Disponível em: https://doi.org/10.1145/3351095.3372843.

PAPINENI, K. et al. Bleu: a method for automatic evaluation of machine translation. In: **Proceedings of the 40th annual meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2002. p. 311–318.

PARTER, S. V.; YOUNGS, J. W. The symmetrization of matrices by diagonal matrices. **J. Math. anal. and Appls.**, Cornell Univ., Ithaca, NY, v. 4, 1962.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543.

PENNY, R.; PENNY, R. J. **A history of the Spanish language**. [S.l.]: Cambridge University Press, 2002.

PETRESKI, D.; HASHIM, I. C. Word embeddings are biased. but whose bias are they reflecting? Springer, v. 38, p. 975–982, 2023.

PILEHVAR, M. T.; CAMACHO-COLLADOS, J. Embeddings in natural language processing: Theory and advances in vector representations of meaning. **Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers, v. 13, n. 4, p. 1–175, 2020.

POSNER, R. et al. The romance languages. Cambridge University Press, 1996.

QI, Y. et al. When and why are pre-trained word embeddings useful for neural machine translation? **arXiv preprint arXiv:1804.06323**, 2018.

RUDER, S. An overview of gradient descent optimization algorithms. **arXiv preprint arXiv:1609.04747**, 2016.

SARDINHA, T. B.; FERREIRA, T. d. L. S. B. **Working with Portuguese corpora**. [S.l.]: A&C Black, 2014.

SCHMID, E. C. Developing competencies for using the interactive whiteboard to implement communicative language teaching in the english as a foreign language classroom. **Technology, Pedagogy and Education**, Taylor & Francis, v. 19, n. 2, p. 159–172, 2010.

SCHÖNEMANN, P. H. A generalized solution of the orthogonal procrustes problem. **Psychometrika**, Springer, v. 31, n. 1, p. 1–10, 1966.

SIMON, C. **The Orthogonal Procrustes problem**. 2018. https://simonensemble.github.io/posts/2018-10-27-orthogonal-procrustes/ (Accessed on May 20, 2022).

SINAP, A.; ASSCHE, W. V. Orthogonal matrix polynomials and applications. **Journal of Computational and Applied Mathematics**, Elsevier, v. 66, n. 1-2, p. 27–52, 1996.

SMITH, D. **How Long Does Translation Take?** 2019. https://gengo.com/business-insights/how-long-does-translation-take/ (Accessed on August 15, 2023).

SOLUTIONS, D. **CBOW representation of the sentence "the cat jumped off the chair"**. 2016. http://deep-solutions.net/blog/WordEmbeddings.html (Accessed on May 15, 2022).

STAHLBERG, F. Neural machine translation: A review. **Journal of Artificial Intelligence Research**, v. 69, p. 343–418, 2020.

TABLE, A. **Ascii table**. 1979.

TITLER, M. G. Translation research in practice: an introduction. **Online Journal of Issues in Nursing**, v. 23, n. 2, 2018.

UPADHYAY, S. et al. Cross-lingual models of word embeddings: An empirical comparison. **arXiv preprint arXiv:1604.00425**, 2016.

VYAS, Y.; CARPUAT, M. Sparse bilingual word representations for cross-lingual lexical entailment. In: **Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies**. [S.l.: s.n.], 2016. p. 1187–1197.

WANG, H. et al. Progress in machine translation. **Engineering**, Elsevier, 2021.

WOOLLEY, G.; WOOLLEY, G. **Reading comprehension**. [S.l.]: Springer, 2011.

XIA, P.; ZHANG, L.; LI, F. Learning similarity with cosine similarity ensemble. **Information Sciences**, Elsevier, v. 307, p. 39–52, 2015.

XING, C. et al. Normalized word embedding and orthogonal transform for bilingual word translation. In: **Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies**. [S.l.: s.n.], 2015. p. 1006–1011.

YANG, W.; OGATA, J. Aistai neural machine translation systems for wat 2019. **WAT 2019**, p. 159, 2019.

YIN, Z.; SHEN, Y. On the dimensionality of word embedding. **Advances in neural information processing systems**, v. 31, 2018.

ZENS, R.; OCH, F. J.; NEY, H. Phrase-based statistical machine translation. In: SPRINGER. **KI 2002: Advances in Artificial Intelligence: 25th Annual German Conference on AI, KI 2002 Aachen, Germany, September 16–20, 2002 Proceedings 25**. [S.l.], 2002. p. 18–32.

ZHAO, J.; SARKAR, V. Intermediate language extensions for parallelism. In: **Proceedings of the compilation of the co-located workshops on DSM'11, TMC'11, AGERE! 2011, AOOPES'11, NEAT'11, & VMIL'11**. [S.l.: s.n.], 2011. p. 329–340.

ZHAO, J. et al. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. **arXiv preprint arXiv:1707.09457**, 2017.

ZOU, W. Y. et al. Bilingual word embeddings for phrase-based machine translation. In: **Proceedings of the 2013 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2013. p. 1393–1398.