UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Study and automatic translation of a language with limited resources

End of Degree Project

Bachelor's Degree in Informatics Engineering

AUTHOR: Baggetto Chamero, José Pablo

Tutor: López Rodríguez, Damián

Experimental director: LARRIBA FLOR, ANTONIO MANUEL

ACADEMIC YEAR: 2021/2022

Resum

El problema de la traducció automàtica és particularment interessant en aquells idiomes per als quals, per qüestions de població, rellevància política, aïllament o altres motius, la quantitat de recursos disponibles són limitats, on per recursos habitualment s'entenen traduccions existents de, o des de, l'idioma a qualsevol altre amb una situació més avantatjosa. Actualment, aquest problema és interessant perquè països en aquestes circumstàncies pertanyen a institucions internacionals on es planteja la traducció automàtica com a solució a les reunions multilaterals. És un problema especialment important al paradigma actual, ja que l'estat de l'art el constitueixen models estadístics que necessiten grans volums de dades per poder aprendre els patrons subjacents del llenguatge. En aquest TFG es planteja considerar un idioma artificial, proposat formalment el 2014, el Toki Pona com a banc de prova que abordi el problema anteriorment exposat. Aquest idioma minimalista cerca expressar el màxim de significats amb una complexitat mínima. Si bé Toki Pona no es planteja, al contrari de l'esperanto, com una llengua de comunicació internacional, sí que comparteix amb ell característiques com la senzillesa d'aprenentatge. Des de la seva presentació informal a la web el 2001, la comunitat Toki Pona ha crescut en nombre i activitat, i hi ha recursos amb traduccions generats per aquesta que s'utilitzaran com a base per a l'aprenentatge d'una xarxa neuronal per a la traducció automàtica. Aquests recursos estan avalats per la comunitat, per la qual cosa es poden considerar dades fiables en un procés d'aprenentatge automàtic. Una experimentació obtindrà els resultats que permetin obtenir conclusions i possibles línies de treball. A més, el TFG planteja el disseny d'una gramàtica formal, no existent fins ara, que permeti estudis futurs utilitzant tècniques diferents de l'aprenentatge de xarxes neuronals.

Paraules clau: Traducció Automàtica, Toki Pona; Xarxes neuronals; Aprenentatge automàtic.

Resumen

El problema de la traducción automática es particularmente interesante en aquellos idiomas para los que, por cuestiones de población, relevancia política, aislamiento u otros motivos, la cantidad de recursos disponibles son limitados, donde por recursos habitualmente se entienden traducciones existentes de, o desde, el idioma a cualquier otro con situación más ventajosa. Actualmente, este problema es de interés dado que países en estas circunstancias pertenecen a instituciones internacionales donde se plantea la traducción automática como solución a las reuniones multilaterales. Es un problema especialmente importante en el paradigma actual, ya que el estado del arte lo constituyen modelos estadísticos que necesitan de grandes volúmenes de datos para poder aprender los patrones subyacentes del lenguaje. En este TFG se plantea considerar un idioma artificial, propuesto formalmente en 2014, el Toki Pona como banco de prueba que aborde el problema anteriormente expuesto. Este idioma minimalista busca expresar el máximo de significados con una complejidad mínima. Si bien Toki Pona no se plantea, al contrario del Esperanto, como una lengua de comunicación internacional, sí comparte con él características como la sencillez de aprendizaje. Desde su presentación informal en la web en 2001, la comunidad Toki Pona ha crecido en número y actividad, existiendo recursos con traducciones generados por esta que se utilizarán como base para el aprendizaje de una red neuronal para la traducción automática. Estos recursos están avalados por la comunidad por lo que pueden considerarse datos fiables en un proceso de aprendizaje automático. Una experimentación obtendrá los resultados que permitan obtener conclusiones y posibles lineas de trabajo. Además, el TFG plantea el diseño de una gramática formal, no existente hasta el momento, que permita estudios futuros utilizando técnicas distintas al aprendizaje de redes neuronales.

Palabras clave: Traducción automática; Toki Pona; Redes neuronales; Aprendizaje automático.

Abstract

The problem of machine translation is particularly interesting in those languages for which, due to population, political relevance, isolation or other reasons, the amount of available resources is limited, where resources usually mean existing translations from or into any other language with a more advantageous situation. This problem is of current interest given that countries in these circumstances belong to international institutions where machine translation is proposed as a solution to multilateral meetings. It is a particularly important problem in the current paradigm, since the state of the art is constituted by statistical models that require large volumes of data to learn the underlying patterns of the language. In this TFG we propose to consider an artificial language, formally proposed in 2014, Toki Pona as a testbed that addresses the above problem. This minimalist language seeks to express the maximum number of meanings with minimal complexity. While Toki Pona is not intended, unlike Esperanto, as a language for international communication, it does share with it features such as simplicity of learning. Since its informal presentation on the web in 2001, the Toki Pona community has grown in number and activity, and there are resources with translations generated by it that will be used as a basis for learning a neural network for automatic translation. These resources are endorsed by the community so they can be considered reliable data in a machine learning process. An experimentation will obtain the results that will allow conclusions and possible lines of work to be drawn. In addition, the TFG proposes the design of a formal grammar, not existing so far, which will allow future studies using techniques other than neural network learning.

Key words: Machine Translation, Toki Pona; Neural networks, Machine learning.

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

Deep learning, and especially transformers, have provided high-quality translations that would be hard to imagine ten years ago. The downside of these advancements is that lots of data must be provided to the model in order to obtain high-quality translations. For this reason, languages with few resources do not have access to these tools, this is the case for Toki Pona.

Toki Pona is a minimalistic synthetic language which is spoken among some online communities. This language is constructed to have a small vocabulary and simple grammar. For this reason, Toki Pona is one of the most ambiguous languages given its high degree of contextuality. Therefore the translation tools are limited.

This work aims to provide a good quality translator for the Toki Pona community and to study and overcome the problems of machine translation in languages with limited resources.

1.1 Motivation

In recent years, there has been a revolution in the NLP field thanks to the popularization and development of deep learning in this area. Thanks to this phenomenon, people with access to the internet can have high-quality translations instantly for the most widely spoken languages in the world. In parallel, languages with very few speakers and not many texts written in those languages cannot profit from these advances as deep learning technology requires large amounts of data.

In this problem, we wanted to focus our work in Toki Pona for several reasons. First of all, Toki Pona is a language with very few speakers which make use of the language in niche online communities. This is very helpful as the speakers can easily be contacted and are invested in the language.

Secondly, Toki Pona is a minimalistic language which heavily relies the meaning of the sentences on its context. This is a very interesting property and we wanted to explore its benefits and its problems. For this reason, we developed a formal grammar and grammatical tools that take advantage of the simple nature of the rules in the language. In contrast, we also wanted to develop a machine translator, where meaning is very important and the ambiguity of the sentences can damage severely the performance of the model and the posterior evaluation.

Moreover, Toki Pona is an artificial language which was created with a unique grammar and does not share many similarities with the most common languages. For this reason, we wanted to explore the benefits of transfer learning in languages that are that distinct as it is a common practice for many languages for small resources.

For all these reasons, this study is valuable not only by providing useful resources for the Toki Pona community, but also supposes a complete revision on the problems that may arise with minor languages.

1.2 Objectives

The main goal of this work is to develop a functional automatic translator for the Toki Pona community. For this purpose, the following specific objectives are addressed:

- Study and creation of the formal grammar for Toki Pona.
- Develop grammatical tools (grammar checker and sentence generator) for the language.
- Make a recompilation of sentences for the creation of a dataset for the Toki Pona - English translation.
- Create a Toki Pona translator from scratch and fine-tune its parameters to obtain a better performance.
- Development of an English to Spanish translator using large amounts of data.
- Make use of transfer learning to learn the Spanish model Toki Pona.
- Compare both models in order to study which is the most beneficial technique and obtain the best model.
- Open-source the results of this work so that the Toki Pona community can benefit and build on top of the results here presented.

1.3 Structure

The structure of this work will consist firstly of a revision on the Toki Pona language and the technologies, algorithms, techniques and metrics that make state-of-the-art translations possible.

In the following chapter, it is explained all our work related to the construction of a formal grammar for Toki Pona and the tools developed based on it. These tools are a grammar checker and a sentence generation. We will also explain how we used them to validate each other.

The following chapter is entirely dedicated to the creation of the two Toki Pona translators that we built to be compared. One which is going to be trained from scratch and another one making use of transfer learning techniques.

Finally, this report ends with the conclusions and ideas for future development and advance of this work.

CHAPTER 2
General context

2.1 Toki Pona

If English is a thick novel, then
Toki Pona is a haiku.

Sonja Lang

Toki Pona is a philosophical and artistic artificial language created by the Canadian linguist and translator Sonja Lang. The first drafts of the language were published on the internet in 2001 and since then, an increasingly growing community of online speakers emerged fascinated by this language. Later, in 2014, she published *Toki Pona: The Language of Good*[5] where she expressed the grammar rules and the official list of the vocabulary. Recently, in 2021, she has published a dictionary called Toki Pona Dictionary, based on the community usage of the language[6]. In 2021, a census was created by the Toki Pona community. Around 1000 people answered the poll, where more than 650 people claimed to know Toki Pona and 165 said they were advanced or fluent speakers.

Toki Pona is lexically, phonetically and semantically minimalist. It uses the minimum amount of elements and the simplest ones in order to express as much as possible. That is why the language is constructed with only around 120 words and 14 letters of the alphabet.

In contrast with other languages, Toki Pona breaks down ideas into their most basic elements. For example, a geologist is the same as a "person of earth knowledge" and to be hungry is the same as wanting to eat. Moreover, there exists many synonymous words like "big", "large" or "huge" which have a very slight difference between them and are not necessary in many situations. In order to simplify these situations, each word has been carefully selected to cover a broad range of meanings. For example, *lipu* is any document, book, postcard or even a clay tablet as it expresses any flat object.

As a consequence of its minimalist approach, Toki Pona often lacks the ability to distinguish finer shades of meaning, being too general and vague. For example, by grouping every bird under the word *waso*, we eliminate the need to learn hundreds of vocabulary items since we are grouping every single species of bird

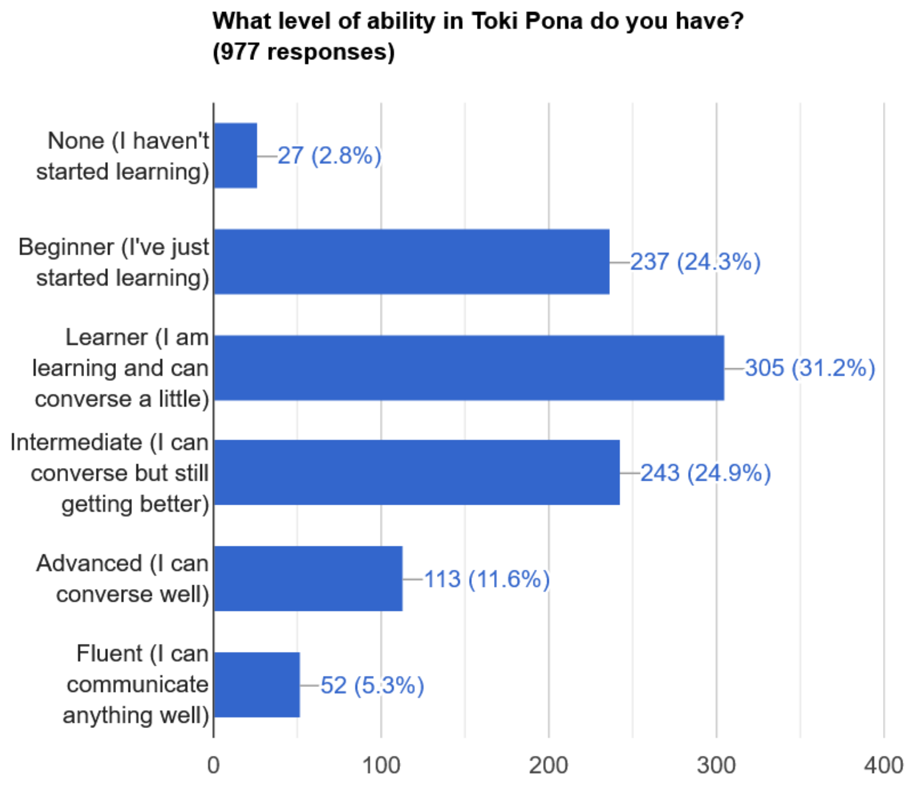

Figure 2.1: Level of speakers of Toki Pona that answered in English

in a word. However, we lack the capacity to distinguish between chickens and eagles. We can approximate this by saying "stupid bird" *waso nasa* and "strong bird" *waso wawa*. Furthermore, Toki Pona is useful to communicate about your feelings or everyday activities. However, it is almost impossible to translate a chemistry textbook or a legal document without losing significant meaning.

These characteristics and limitations of Toki Pona are deliberately designed, as Toki Pona is created with an ideology based on the benefits of living a simple life and this language suits this lifestyle perfectly. For this reason, the language is called Toki Pona, which means 'good language' or 'simple language' as the word *pona* means good and simple, which is equivalent for the author.

All these peculiarities of Toki Pona have been a challenge when trying to create a translator. This is caused by the enormous ambiguity of the language which is heavily influenced by context, since the words have multiple meanings and they are very general. However, the minimalistic approach has been a huge advantage developing a formal grammar for the language, as it is much shorter and simpler that any of the most frequently spoken languages.

Another thing to keep in mind when working with Toki Pona is that it has many writing systems. Toki pona can be represented with letters and characters used in other spoken languages. In this category, some of the most popular systems are the Chinese, Arabic, Hebrew or Latin, which is the most used system. Latin is the preferred system in chats and web pages because it has UNICODE

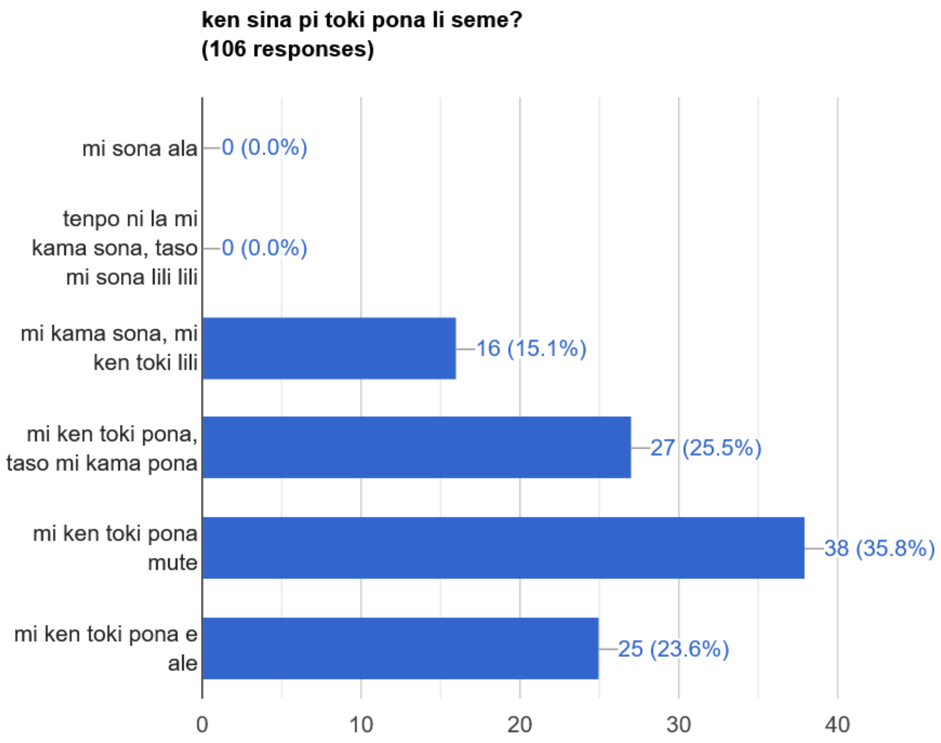

Figure 2.2: Level of speakers of Toki Pona that answered in Toki Pona

representation and most speakers write with this system in their native language or have studied a language where they do.

However, the favourite system by the community when these limitations are nonexistent is the *sitelen pona*, which is an ideographic system where each character or glyph represents a word created by Sonja Lang. Another popular system is *sitelen sitelen*, a more complex system that looks similar to Mayan writings. It was created by Jonathan Gabel and introduced in *Toki Pona: The language of good*[5]. An example of these writing systems can be seen in Figure 2.3. For simplicity, we only considered the Latin alphabet for this work.

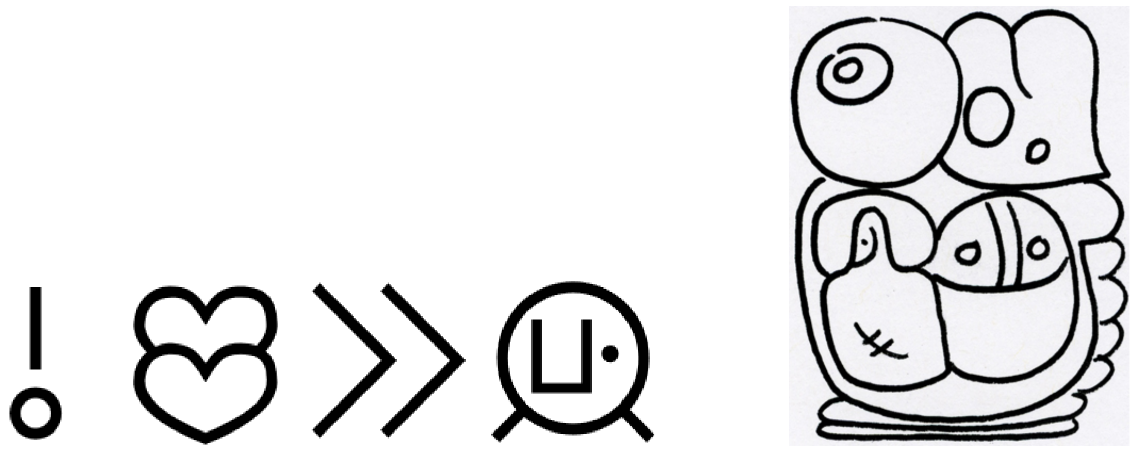

Figure 2.3: *o olin e jan poka*, which means "love your neighbour", written in *sitelen pona* and in *sitelen sitelen*

2.2 Formal Grammars

A formal grammar is G is defined by the tuple (N, Σ, P, S) where each component means:

- N: A finite set of nonterminal symbols disjoint with the strings formed from G.
- Σ: A finite set of terminal symbols disjoint from N. It is also called the alphabet of the grammar.
- P: A set of production rules that follow the form $(\Sigma \cup N)^* N (\Sigma \cup N)^* \rightarrow (\Sigma \cup N)^*$ where $(\Sigma \cup N)^*$ represents any string that is formed by terminal and nonterminal characters, including the empty string.
- S: A symbol $S \in N$ that represents the starting symbol.

In a grammar, a state is a string that can be reached when applying a rule to another state and the starting symbol. When the string is entirely composed of terminal symbols, this state is called terminal or word. The language L described by the grammar is the set of words that can be obtained from the start symbol using the productions in the grammar.

For example, a grammar

$$G = \{N = \{S, A\}, \Sigma = \{a, b\}, P = \{\{S \rightarrow aA\}, \{aA \rightarrow b\}, \{A \rightarrow a\}\}, S\}$$

can be derived like $S \rightarrow aA \rightarrow aa$ or like $S \rightarrow aA \rightarrow b$. Therefore, the language described by the grammar G is L={aa, b}.

2.2.1. Context-free grammars

Context-free grammars are a class of formal grammars where the productions P have the form $N \rightarrow (\Sigma \cup N)^*$. In other words, the left-side of the production must be a single nonterminal symbol.

This type of grammars have great importance in linguistics as it is believed that many natural languages belong to this group. In addition, most of the programming languages are designed to be context-free languages and are defined by a context-free grammar. This is done to benefit from the parsing algorithms available for this class of languages.

Every context-free grammar can be converted into Chomsky normal form (CNF). This is a grammar where every rule must follow one of the following structures:

$$A \rightarrow BC$$

$$A \rightarrow a$$

$$S \rightarrow \varepsilon$$

Where A, B and C are arbitrary nonterminal symbols, S the starting symbol, a is an arbitrary terminal symbol and ε denotes the empty string.

The most important algorithm designed for parsing context-free languages is the Cocke–Younger–Kasami (CYK) algorithm[7]. This algorithm requires the grammar to be in Chomsky Normal Form. The importance of the algorithm resides in the high time efficiency as, using big O notation, the worst case is $\mathcal{O}(n^3|G|)$ being n the length of the parsed string and |G| the size of the grammar.

However, this class of languages has some inconveniences. One of the most well known problems in context-free grammars is the ambiguity problem. A grammar is ambiguous if there exists a string that can have more than one leftmost derivation or rightmost derivation. A leftmost derivation is applying the rules always to the leftmost nonterminal symbol of the string and a rightmost derivation is the same procedure but always applying the rules on the right.

Unfortunately, the problem of determining if a context-free grammar is ambiguous is undecidable. Furthermore, it has been demonstrated that some languages cannot be represented with an unambiguous grammar. This languages are called inherently ambiguous languages.

2.3 Deep Learning

Machine Learning is the section of Artificial Intelligence (AI) which aims to develop software that learns how to solve a problem in an autonomous way, without the need for explicit programming. These models learn to develop their tasks based on the data they obtain. This learning can be classified in different classes depending on the data provided to solve the problem:

- **Supervised Learning:** Data is labeled and the model learns based on the expected output that is desired. This is the most common type of learning.
- **Unsupervised Learning:** In contrast to supervised learning, data is unlabeled and the model learns based on finding underlying patterns in the data. One of the most common problems of this type is clustering, which aims to classify data based on proximity groups.
- **Reinforcement Learning:** This type of learning learns dynamically based on a system of rewards and punishment that makes it select the strategy which gives them the maximum benefit.

There exist many machine learning models (like perceptron[8], decision tree learning[9]...) but the most popular nowadays are neural networks. Their structure is inspired in the neurons that form the nervous system in humans and other animals. This is simulated by creating a big net of interconnected elements that can understand complex realities based on simple elements. In order to achieve that, the net receives a vast quantity of examples (data) that allow the net to learn and adapt in order to acquire the necessary knowledge for the task.

Deep learning is the section of Machine Learning that is based on deep neural network computational architectures. This means that it is composed of several layers of neurons interlinked between them.

The emergence of the theory on which deep learning basis expands between the 1940s (publication of Hebb's rule[10]) and the mid-1980s (introduction of the backpropagation method[11]). However, these techniques have taken more relevance since the late 1990s, early 2000s. This has been due to several factors: advances in the formulation, such as the correction of gradient fading; development of appropriate hardware, where the introduction of GPUs that allow the training of larger architectures in assumable times exploiting the high level of parallelization that these algorithms present and also the availability of datasets of larger sizes stand out.

Nowadays, deep learning has diversified into many tasks and domains. For this reason, there exist different configurations to obtain a better development in those specific areas. Some of the most widespread architectures are:

- **Convolution Neural Network (CNN):[12]** They are a specialized kind of neural network for processing data that has a known grid-like topology. That is why they are very popular for computer vision. The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. This operation consists of multiplying every number for the corresponding number in a kernel and then making the sum of all the multiplications. This operation is done with the kernel acting as a sliding window until the whole destination layer is complete (Figure 2.4).

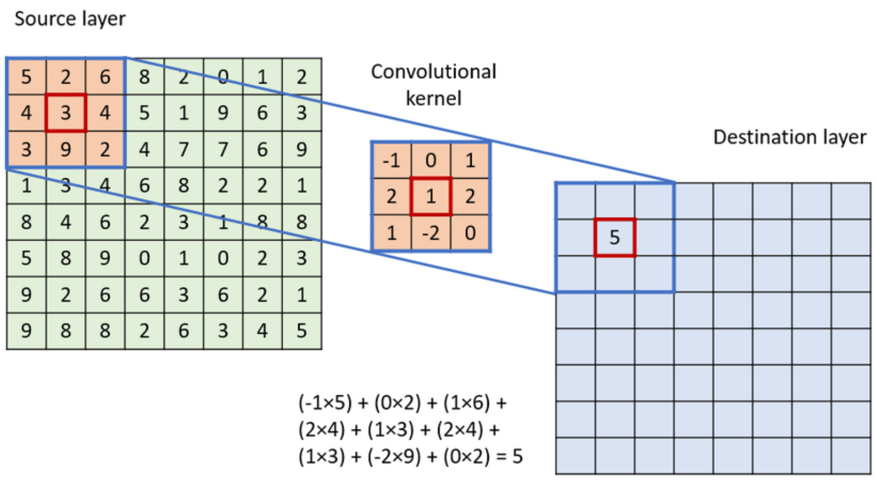

Figure 2.4: Convolution operation [1]

- **Autoencoders[13]:** This architecture is generally used for dimensionality reduction and for generative models. It is designed to attempt to copy its input to its output but approximately and following some restrictions. Because the model is forced to prioritize which aspects of the input should be copied, it often learns useful properties of the data. Its structure consists of an encoder and a decoder.

- **Long Short-Term Memory (LSTM)[14]:** Developed for use on sequential data, they are an improved implementation of Recurrent Neural Networks (RNN). LSTM solves the long term memory problem of RNNs, allowing them to manage inputs with variable and arbitrary length.

- **Transformers[4]:** The transformer architecture is very recent but has dominated since its creation many areas such as Natural Language Processing and some problems in computer vision. They follow the idea behind LSTM but go one step further. They completely remove the recurrent structure and develop an attention system that links the elements of the sequence. These models will be further explained in section 2.5.

One problem with deep learning is the enormous amount of data that is required to train a model. In addition, these models have to be very big and include a large quantity of parameters in order to provide state-of-the-art results. These parameters are randomly initialized and when the number of them increases, more data is needed to make them converge.

Both quantities have been increasing exponentially every year and it is thought that the tendency will remain.

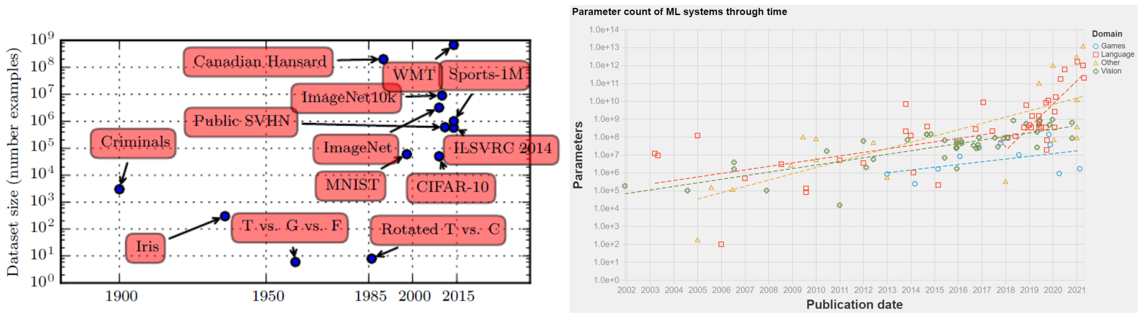

Figure 2.5: Evolution of dataset size[2] and number of parameters in deep learning[3]

For this reason, transfer learning techniques were introduced. These techniques allow big models trained for very used tasks, in general domains and in widely spoken languages (in the case of NLP) to be fine-tuned in order to obtain good results in specific tasks, particular domains and minority languages.

2.3.1. Transfer Learning

Before the apparition of Transfer Learning, models were trained using as much data as possible from the task that they were targeting. Nowadays, models require to be trained with large amounts of data and are composed of billions of parameters. For this reason, state-of-the-art models require to be trained by very expensive and high-performance hardware for months. Furthermore, there does not exists enough data about some specific tasks and data. To solve this problem, Transfer learning was introduced.

In transfer learning models, there exist two phases of training: pre-training and fine-tuning.

- **Pre-training:** In this stage, models are trained with very powerful hardware with an enormous amount of generic data. The resulting model is a big model capable of performing very well the tasks that was trained for. However, it is not as good at specific tasks or domains as it could.

This process is generally made by big tech companies, which are the only entities that can afford the hardware and time to develop this pre-trained models. Most of them are publicly available and can be used by other users to fine-tune them in order to accomplish a specific task.

- **Fine-tuning:** Fine-tuning a model consists in training a model with specific datasets about the task that wants to be performed or in the desired domain. This process is not as computing-power demanding as the pre-training. For this reason, it is possible to fine-tune for small companies and particular researchers, as depending on the model, can be trained with GPUs that are optimized for floating-point arithmetics.

However, in recent years, state-of-the-art models in NLP tend to be enormous models capable of performing every task as researchers discovered that many of these tasks benefit between them.

One example of this is the PaLM model developed by Google[15]. This model is a transformer model with 540 billion parameters. It was evaluated for more than 150 NLP tasks giving state-of-the-art in practically all of them.

2.4 NLP

2.4.1. Introduction

Natural language processing (NLP) is the set of methods for making human language accessible to computers. Since the last decade, NLP has entered in our lives and it is here to stay. There are many examples of it in our daily life like automatic machine translation, spam classification for the email, search engines have adopted these systems to make better searches, etc.

There are many applications for NLP. Some of the most important nowadays are:

- **Text classification:** Text classification is the problem that given a text document we must assign a label from the set of all possible labels. One of the most frequent a popular applications of text classification is sentiment analysis (SA). This consists on determining whether a text is positive, negative or neutral.

- **Information retrieval:** It is the task of finding relevant documents in a collection given a query. The most common example of an IR system is the Google search engine. In addition to textual similarity, this system incorporates other factors such as source relevance or page rank.

- **Machine translation:** One of the most well-known problems in NLP and in Machine Learning in general. It consists in translating automatically documents into another language. Even though it is a very researched problem, it is very complex and machines haven't been able to make translations with the quality of professional human translators.

- **Text summarization:** It condenses the information of a text extracting the most important elements and generating words to express it in a coherent and cohesive way.

- **Text Generation:** This application consists of generating correctly formed text based on an incomplete text or a description. This task has many formats and sub-tasks such as automatic code generation, generation of a document based on an incomplete part or a chat-bot assistant.

Machine learning with deep neural networks (especially transformers) are state of art to solve most of these problems.

2.4.2. Preprocessing

Tokenization

As Neural Networks work with vectors and not with words, in order to work in NLP using this technology some transformation must be done. To do it, the first transformation that should be done is to divide the text into tokens. This process is called tokenization.

The most straightforward tokenization is using each word as a token. This approach is good but most languages use derivation and composition, which could be detrimental if we need to build a vocabulary. With the basic tokenization, the vocabulary would include many variations of the same word, which could increase the size of the vocabulary significantly. For this reason, it could be a good idea to separate the words into their root and the derivations of the word. For example, the word *tokenization* could be separated into *token-* and *-ization*.

In order to archive this kind of tokenization there are different techniques. One of the most popular ones is called byte pair encoding (BPE)[16]. This algorithm was created to compress bytes sequences which had similar endings but was adapted for word segmentation[17]. The BPE algorithm consists in two phases: training and inference.

- **Training:** In this step, the algorithm learns which segments has to merge. For this task, it creates a merging table with the merging rules, which will be the output of the step. First, for every word in the vocabulary, it separates its letters and makes each of them the base elements. Then it counts the pairs of elements in the vocabulary and takes the most frequent. The chosen pair is merged for every word and it is used as an element. The algorithm repeats until we reach the maximum number of merges that we established.

- **Inference:** Here, with the merging table obtained with the training, the algorithm applies the rules in order for every word. To distinguish the complete words with the words that have been segmented, the algorithm adds the special characters @@ when the next token is part of the same word.

Embeddings

After the sentences have been transformed into a sequence of tokens, the next step is to transform them into vectors.

One of the most simple ways to represent each word is with a vector of N dimensions, where N is the vocabulary size, with all zeroes except for a 1 in the dimension that represents that word. This representation is called one-hot encoding. The most obvious problem with one-hot encoding is the memory space that is required when the vocabulary size increases. Furthermore, there is a more important problem with this representation: the vectors do not give any information about the word they represent. One-hot vectors are exactly at the same distance from one another. That means that the dog vector is as close to the cat vector as to the table vector. For all these problems, embeddings are the solution.[18]

Embeddings are vector representations of the words that are based on the words that are usually surrounding them, or, as a famous quote says: *"a word is characterized by the company it keeps"*[19]. In order to archive this, there are many techniques. Some of the most famous ones are *Word2Vec*[20] and *GLoVe*[21]. In deep learning, embeddings are trained with the rest of the model.

2.5 Transformers

2.5.1. Introduction

Recurrent Neural networks and Long Short-Term Memory (LSTM) had been the state-of-the-art approach in sequence modeling and transduction problems like machine translation in which an encoder-decoder architecture is necessary. Despite that, they had some problems that made its performance not as good as it could.

Recurrent architectures as the mentioned before, in theory, should maintain the information along the sequence so it can generate new tokens based on the previous ones. However, in practice, as memory constraints limit batching across the examples, the model tends to forget the first tokens giving undesirable results. In LSTM architectures, some attention mechanisms were introduced to solve this problem. Nevertheless, the problem was inherent to the recurrent architecture. It was in this context that transformers were created[4].

Transformers completely take recursion out of the equation and draw the dependencies between the different elements solely relying on the attention mechanisms. This strategy solved the previous problem and, additionally, increased the parallelization, as the sequential nature of the recurrent architectures disappeared.

2.5.2. Self-Attention

The attention mechanism for the transformers is a self-attention one. This is the main difference between the attention in the LSTM models, which the attention

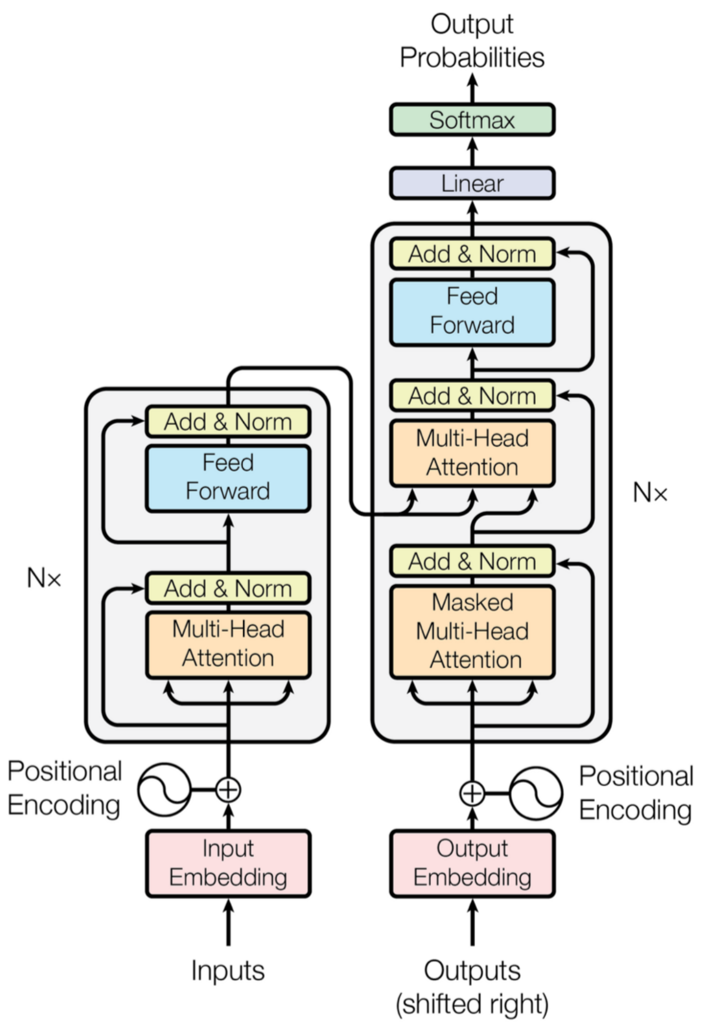

Figure 2.6: Transfomer - Model Architecture[4]

is from one decoder state to an encoder state. In contrast, in transformers, the attention is from each state of a set to the other states of the set.

This is implemented with three vectors associated with each token. One query vector, a key vector and a value vector. The query asks for the information, the key informs that it has the information and the value is the information. This system is achieved with the following formula where Q, K and V are the query, key and value vectors respectively and d_k is the number of dimensions of K.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

In the decoder, the self-attention mechanism is slightly different. Each token can only "see" the tokens that appear before them in the sentence. This is called masked self-attention.

It was tested that models needed to focus on different characteristics in order to be better at deciding which words need to have a higher attention. This is why multi-head attention was created. Multi-head attention calculates the individual attention and concatenates the vectors according to this formula where the Ws are the weights of the model:

$$MultiHead(Q,K,V) = Concat(head_1, ..., head_h)W^O$$
$$\text{where } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

2.6 BLEU

BLEU [22](bilingual evaluation understudy) is a method created in order to evaluate the quality of a machine translation. In order to suit this porpoise, the evaluation is designed to be automatic, quick, language-independent and that corresponds with human evaluation.

The main idea of the algorithm is to determine the quality of the machine translation based on a professional human translation which is used as a reference. The method returns a value from 0 to 1 as the BLEU score of the translation, being 0 the lowest score and 1 the highest. This score is often scaled to range between 0 and 100.

The BLEU algorithm makes its evaluation based on n-gram precision. This means that human and machine translations are divided in groups of n consecutive words called n-grams. Then, the number of n-grams on the machine translation that also appear in the reference (that are clipped) are counted and divided by the total of n-grams in the evaluated sentence. As we usually pretend to evaluate whole documents or corpora and not just a single sentence, all counts are summed. The following formula represents this calculation:

$$p_n = \frac{\sum_{\mathcal{C} \in \{Candidates\}} \sum_{n-gram \in \mathcal{C}} Count_{clip}(n-gram)}{\sum_{\mathcal{C}' \in \{Candidates\}} \sum_{n-gram' \in \mathcal{C}'} Count(n-gram')}$$

In the original paper, researchers found that the ideal number of n for the n-grams ranged from 1 (unigram) to 4 as higher ns were not useful at distinguishing bad from good translations.

As we said earlier, BLEU is based on a precision metric. This has the problem that it does not take into account the recall if no modification was applied to this strategy. This can be seen with the following example:

Human translation: All my friends talk in Toki Pona.

Machine translation: All my friends talk.

Every n-gram in the machine translation is in the reference so the punctuation should be 1 even though the translation is very incomplete. This problem is solved by adding a Brevity Penalty (BP). It is calculated following the next formula where r is the length of the reference and c is the length of the candidate sentence.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

After all these considerations, the formula that calculates the BLEU combines all the n-grams precisions with the geometric mean and applies the Brevity penalty according to the following formula where w_n is $1/4$ as we consider a maximum of 4 n-grams.

$$\text{BLEU} = BP * \exp(\sum_{n=1}^{N} w_n \log p_n)$$

CHAPTER 3
Grammatical tools for Toki Pona

3.1 A formal grammar for Toki Pona

Attempts to construct a formal grammar for Toki Pona have been scarce and not well documented. For this reason, we decided to approach it by constructing a formal context-free grammar based exclusively on the rules in *Toki Pona: The Language of Good*[5].

These rules are dedicated to teach in simple lessons the language. In order to do so, some rules are simplified at the beginning of the book and later generalized and some of them are vaguely described. For this reason, the formal grammar rules and the rules described in the book are different in structure but define the same language.

Furthermore, the grammatical categories described in the book have been reduced. This was the case because in Toki Pona the difference between a verb, a name, an adverb and an adjective are purely semantic and are not reflected in the formal grammar. This simplified the grammar considerably without inconveniences.

One major inconvenience with Toki Pona grammar was the complexity of the question form. In Toki Pona, one way of asking a question is to replace what is unknown with *seme*, and can only appear once. This rule makes it much more complex to find an context-free grammar for the language so it was restrained to the most common uses.

Another problem was the incorporation of foreign words. Foreign words are any words that are not in the original vocabulary of Toki Pona. These could be proper nouns, languages, religions, etc. These words are not possible to be included in the grammar, so there exists a nonterminal state that includes them.

Probably the most significant problem with this grammar is that the language is constantly evolving and there exist different dialects that diverge from the source rules so it is impossible to describe a grammar for this language.

3.2 Grammar checker

After the construction of the context-free grammar, we thought that it would be interesting to create a parser which is able to check if a sentence is in Toki Pona or if it is not. We did it using the Cocke–Younger–Kasami (CYK) Algorithm[7]. This algorithm takes as input a context-free grammar and a word (in this case sentence) and checks if it is in the language described by the grammar.

As it is a well-known algorithm, there are multiple implementations online, so we used Iker García Ferrero's implementation[23]. This implementation has been chosen because it is coded in python and it is easy to use. In order to use this implementation, the grammar must be in Chomsky Normal Form (CNF). To adapt the context-free grammar to CNF there exists another well-known algorithm.

There exists another Toki Pona grammar checker available in the most popular web page of the Toki Pona community designed by Jan Mato. This tool is decent but has some problems. For example, if the parser is given a sentence with the form `Subject "o" Predicate` the parser does not accept it nor reject it, throwing an error.

3.3 Sentence generator

Having the context-free grammar as a base, a sentence generator was created. To develop this generator, we created a program that starts with the label that indicates the start of the sentence and each iteration substitutes the leftmost non terminal label using a rule in the grammar.

Each iteration, the program searches from left to right a non-terminal. When one is found, it searches every rule where the non-terminal appears on the left side of the rule. Then randomly decides which of the rules is going to be used. When the rule has been chosen, it substitutes on the sentence the non-terminal with the right side of the chosen rule and the iteration ends. When the program doesn't find any non-terminals, it ends the execution.

A first version of the program decided which rule was going to be used randomly with each of the candidates having the same probability. This caused that is most cases the algorithm did not converge. To solve this problem, in the non-terminals which were the most problematic, the probability of the rules that led to terminals was increased significantly.

3.4 Validation

To check the correctness of the grammar, we tested it in two different ways. First focusing on the precision and later on the recall.

3.4.1. Precision

When we talk about precision, what is meant is the acceptance of only those sentences that are correct in Toki Pona. To test this, we checked if every sentence of the examples written in *Toki Pona: The Language of Good*[5] was accepted. To make compatible the sentences with the grammar, we had to add to the list of foreign words every one of them that was included in the corpus.

After running the experiment, we realized that some grammar rules were wrong and the grammar was partially rewritten until the test was passed.

3.4.2. Recall

When we talk about recall, what is meant is the acceptance of every Toki Pona sentence that is correct. This was tested generating hundreds of sentences with the Toki Pona generator and using Jan Mato's parser to check if they were accepted.

As we knew the parser wasn't perfect, we asked some members of the community about the errors detected by the parser to ensure if they were truly mistakes of the generator or a mistake of their parser.

After the test was performed, we encountered new mistakes in the grammar, which were corrected and tested again.

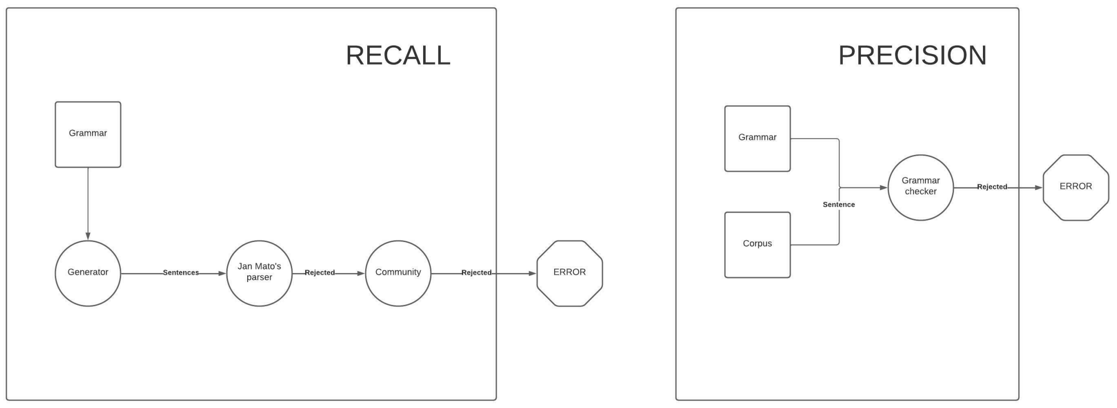

Figure 3.1: Structure of the validation

After the validation every error was corrected. Every sentence in the corpus is accepted by the parser and every sentence generated was grammatically correct.

CHAPTER 4
Machine translation for Toki Pona

In this chapter, we explore different approximations using transformers to develop a good Toki Pona machine translator. For this task, we required a GPU to perform all the calculations needed for the training of the network. The GPU used in this work is an NVIDIA GeForce GTX 1060 with 6 GB of GPU RAM. Unfortunately, this is not a good GPU to work with mainly for the RAM limitation making large models impossible to be trained.

4.1 Dataset

The main problem in building a Toki Pona translator is the lack of available translations to train the model. Fortunately, this language has an active community that uploads translations made by the speakers to `https://tatoeba.org`, an online collection of sentences and translations. We downloaded from there every pair of sentences that were translated between Toki Pona and English.

All these sentences were divided into three different groups: one for training, another one for validation and a final set for testing. We decided to apply a series of transformations and filtrations of these datasets in order to improve the performance of the training and simplify the task that was going to be performed:

1. Removal of the repeated sentences
2. Removal of every punctuation mark
3. Transformation of every letter into lowercase
4. Removal of every sentence which has a length greater than 35 characters.
5. Random shuffle of the pairs of sentences

After the transformations, we ended up with 15579 pairs for training, 2005 for validation and 1004 for testing.

4.2 Description of the experiment

In order to create a good translator for Toki Pona to English, we decided that we were going to build two different models with transformers. The first one was going to be built from scratch, trying to optimize all the hyperparameters to get the best results.

To build the second model, we wanted to benefit from transfer learning. To do so, we trained a bigger model using a large dataset for the task of translating from Spanish to English. This was the case because it was a language with many speakers and therefore, with many translations available. Once we had this model, we tried to apply fine-tuning giving the model the Toki Pona - English dataset in order to leverage the similarity, if it existed, between Toki Pona and English.

We also explored training the model from other pretrained public models of different languages as they perform significantly better than our first model for being bigger and more trained.

After building these models, we have compared them and chosen the best one.

4.3 Model from scratch

After obtaining the dataset, the next part was to create a model and explore the results changing the hyperparameters. Obtaining good hyperparameters for the model was something crucial in order to obtain a decent result. This is due to the fact that we are using transformers and this technology is very sensitive to them so the difference between good and bad hyperparameters can result in a pretty decent translator or one that translates every sentence in the same way. To make these models (and the ones in section 4.4). we used the library OpenNMT-py.

Some of the parameters have been fixed and have not been explored:

Hyperparameter	Value
accum_count	8
optim	adam
adam_beta1	0.9
adam_beta2	0.998
decay_method	noam
learning_rate	2.0
max_grad_norm	0.0
batch_size	128
batch_type	sents
normalization	sents
label_smoothing	0.1

Table 4.1: Base unexplored parameters for the model from scratch

The most important of these parameters is the optimizer. The optimizer is an algorithm that changes the learning rate in order to make the training more efficient. Adam was chosen because it is usually the one that gives better results.

In addition, the number of steps was decided to be 1000 train steps and to validate and save the current model every 100 steps. If the model started to decrease its performance in validation due to the overtraining the model chosen was the last one before the decay in the accuracy.

The hyperparameters that were explored are:

- Word vector size
- Number of heads
- Size of hidden transformer feed-forward (transformer_ff)
- Number of layers
- Dropout

In addition, it was also tested if a byte-pair encoding preprocess would be beneficial for the model.

As there are many parameters, to explore the best combination we created a base model with standard parameters and changed one or two parameters each time to see if it could improve the model. As it requires much computational power and time, this was the best strategy that we could use based on our limitations.

Furthermore, to test if the bpe encoding preprocessing was better for the model, we tested it against the base model and checked if it increased the score.

The base parameters are:

Hyperparameter	Value
word_vec_size	128
layers	2
transformer_ff	512
heads	2
dropout	0.1

Table 4.2: Base configuration of the model

The results of this exploration are reflected in the next table:

The table shows that the base model has a BLEU score of 20.33, which has been surpassed only by the 3 layers version with 21.07. It can also be seen that the bpe made much worse the translator, for this reason this preprocessing was discarded from the experiment.

In order to try to build an even better model, we combined the version with a dropout of 0.2 (with 19.32 BLEU) and the one with the 3 layers as they are the two features that gave the best result. Unfortunately, the result of this combination

Configuration	BLEU
BASE	20.33
4 HEADS	15.80
8 HEADS	14.80
BPE	8.49
3 LAYERS	21.07
4 LAYERS	18.44
256 WVS	15.29
0.05 DROPOUT	15.61
0.2 DROPOUT	19.32
FF 1024	15.79
FF 1024 4 HEADS	13.60
3 LAYERS 0.2 DP	18.33

Table 4.3: BLEU of the different configurations of hyperparameters

had a score of 18.33, which did not provide a better score that the other versions alone.

We obtained a best BLEU score of 21.07. This score is probably lower than it should because Toki Pona is an ambiguous language and a sentence can have hundreds of translations. This is due to the fact that Toki Pona is a minimalist language that heavily relies on context. That means that the same sentence can have two different meanings or define a more or less specific situation. This causes that some translations that are correct get a very low BLEU score because the original sentence meant a completely different thing.

Some examples of translations where this effect occurs:

Source:	mi jo ala e telo nasa
Human:	i don t have vodka
Machine:	i don t have any wine
Here the type of alcohol is not specified and depends on the context.	

Source:	sina pilin ala pilin e ni jan sewi li lon
Human:	do you believe that god exists
Machine:	do you believe in god
The two translations mean the same but the structure is different.	

Source:	jan ton li wile utala
Human:	tom is aggressive
Machine:	tom wants to fight
It is not specified if Tom generally wants to fight (is aggressive) or if he wants to fight now.	

Source:	mi sona ala e nimi ona
Human:	i don t understand her words
Machine:	i don t know her name
Both translations are correct as *sona* means to understand or to know and *nimi* means word or name.	

4.4 Our Transfer Learning model

Once we obtained the first model and thought about the primary limitation on training the model, the low number of translations, we decided that we could build a better model that tries to solve this issue. The approach was to build a model from a language that had a large dataset of translations and resembled Toki Pona. Unfortunately, according to the Toki Pona community, the two languages that are more similar to Toki Pona are Tok Pisin and Esperanto, which neither of them is largely spoken.

Facing this situation, we decided to try using Spanish since it is a language we are familiar with and widely used around the world.

To train this model, we used the Europarl parallel corpus [24], a widely used and known dataset in NLP and Machine translation which is extracted from the proceedings of the European Parliament. This dataset is translated to 21 different European languages and some languages like Spanish have almost two million sentences for training (1,965,734) translated to English, which made this corpus very appropriate for our task.

When we had all data downloaded, we had 1,965,734 pairs of sentences, divided into 3 sets: the training set, which contained 1,965,734; the test set, with 3000 pairs of sentences, and the development test with 3003 pairs. These sets were transformed according to the steps in Chapter 4.1. When the transformations are applied, the dataset contains respectively 1,410,013; 3000; 3002 sentences of pairs.

After this, we had to choose the hyperparameters to train the model. As explained before, these parameters are very important and the first models that we created could not learn how to translate anything and had results of around 2 points of BLEU. This happened because the models that we were trying to build were very big and had many parameters. As the training set was not big enough for all the parameters we suffered from the "curse of dimensionality" as the big number of parameters in the model made that in the multidimensional space that represent it, the data are very sparse, and therefore, much more data is needed to create a good generalization.

The models trained for a maximum of 176250 train steps with a batch size of 16. That makes for a maximum of 2 epochs. When the two epochs were completed the model that had a greater accuracy score with the development test was picked and evaluated with the test score.

Testing smaller models, we obtained one that gave a BLEU score of 18.55 with the following configuration:

Hyperparameter	Value
word_vec_size	128
layers	2
transformer_ff	128
heads	1
dropout	0.1

Table 4.4: First configuration of the Spanish to English model

Seeing that this configuration managed to make a decent result, we tried to change the parameters slightly to obtain a better configuration until we got our maximum of 23 BLEU with the following configuration:

Hyperparameter	Value
word_vec_size	256
layers	4
transformer_ff	256
heads	4
dropout	0.1

Table 4.5: Final configuration of the Spanish to English model

After reaching this mark we decided that it was a good enough score as the original paper of the dataset [24] is 30 BLEU we thought we were quite close and if we wanted a better score we would need to implement embeddings or other complex techniques that are out of the scope of this work.

Once we obtained the Spanish - English model we could return to our original task: create the Toki Pona - English translator. To build it we trained the model with the same parameters except for the learning rate. In order to obtain a good model, two more tests were created. Trying to obtain a good learning rate that could learn Toki Pona but did not forget much Spanish. On the same line, we made experiments retraining the model with a pondered combination of the Spanish - English and the Toki Pona - English corpora and the Toki Pona - English corpus alone.

The learning rates that we tried were 2 (the original) , 0.1 and 0.01. These two in practice performed exactly in the same way because in opennmt-py, the NOAM optimizer has a maximum real learning rate of 0.00042, which both of the values get for every iteration of the training.

The combination of corpora was merged with a weight of 0.8 for the Toki Pona corpus and 0.2 for the Spanish corpus.

Unfortunately, none of the experiments created a translator which made good translations in Toki Pona. This might be due to the fact that the created model was not built with enough parameters and the sufficient complexity that is required to slightly alter the state of the model in order to benefit from transfer learning.

Nonetheless, we believe that transfer learning is a powerful technique that can benefit low-resource languages. For this reason, we decided to make another experiment using big and public pretrained models in different languages in order to try to solve the problems encountered with Transfer Learning. Furthermore, we wanted to fine-tune from different languages in order to find out which one benefits Toki Pona more.

4.5 OPUS Transfer Learning

For this section, we wanted to explore if the Transfer Learning technique could benefit our model if it was trained in bigger models. For this reason, it was decided to change the library used to one with more facilities for fine-tuning public models. The library that fulfilled our requirements was HuggingFace.

HuggingFace is a deep learning library dedicated to transformers and mainly focused on NLP. The library runs over the most commonly used deep learning frameworks Pytorch and TensorFlow. Pytorch was the option chosen for this investigation. Furthermore, it is a repository of pretrained models and datasets that can be very easily downloaded and used.

The pretrained models used for the experiment are the OPUS models developed by Jörg Tiedemann[25]. These models had been trained using the transformer model in the MarianNMT framework[26]. The data used to train the model was obtained from the OPUS dataset[27].

The languages chosen for our investigation are Spanish (es), French (fr), German (de), Russian (ru), Arabic (ar), Japanese (ja), Chinese (zh), Esperanto (eo) and Tok Pisin (tpi). These languages were chosen because they were among the languages with the higher number of training examples or, in the case of Esperanto and Tok Pisin, because they have been the inspiration for some words in the vocabulary and we thought it could be interesting to explore them. The next table shows the number of sentences which have been used for pretraining each model:

Language	Sentences
eo	19.4M
tpi	405k
fr	479.1M
de	349.0M
ja	68.1M
ru	213.8M
ar	102.8M
es	553.1M
zh	103.2M

Table 4.6: Number of sentences used to pretrain each translation model

In addition, the model trained has the following hyperparameters:

Hyperparameter	Value
word_vec_size	64
layers	6
transformer_ff	2048
heads	8
dropout	0.1

Table 4.7: hyperparameters of the pretrained models

This configuration of the network is way bigger than the one used in section 4.4. Training models of this size is an impossible task with the GPU which we had available as it has a limited GPU RAM memory, or, it would have to be trained with a very small batch size, resulting in a very long training (Our estimation is that it would take approximately three years for the Spanish model) as many of this languages have almost 500 times more training examples that the ones we used.

The pretrained models were finetuned with our toki pona dataset with a batch size of 8 sentences. Each model was trained for eight epochs. In each epoch, the models were evaluated with the development dataset and a checkpoint was saved.(Fig. 4.1)

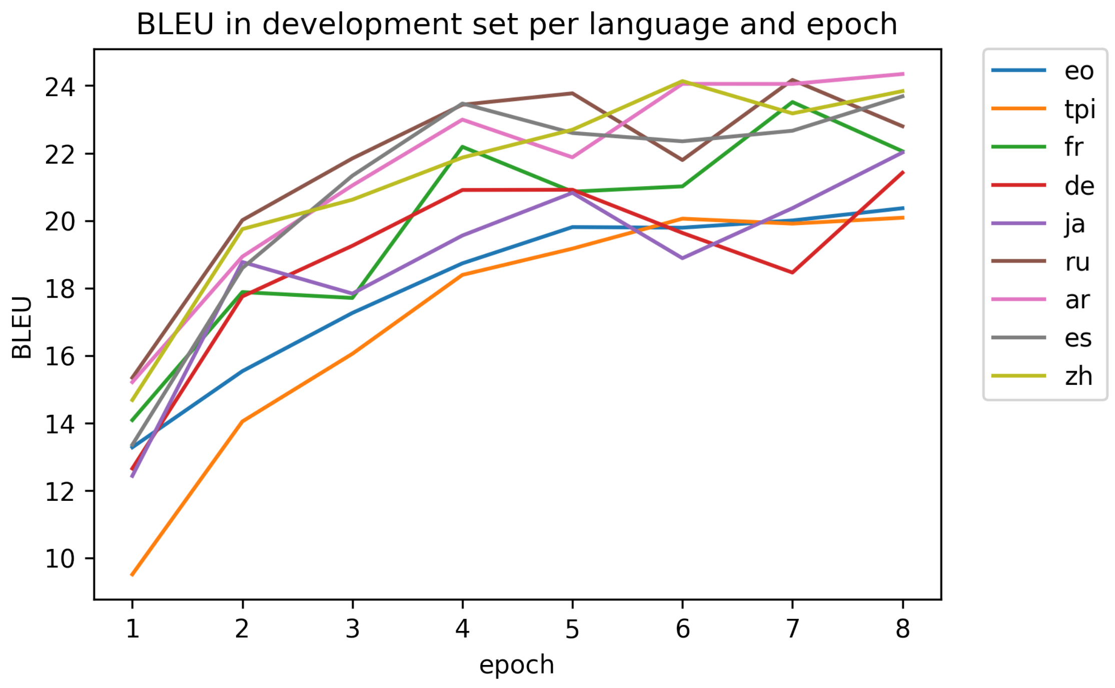

Figure 4.1: BLEU in the development set for each epoch

After the training, for each language, the checkpoint of the epoch that obtained the best evaluation BLEU was evaluated again but with the test set. Results compared with the model trained from scratch in Fig. 4.2.

In the graph, it can be seen that some models (Arabic, Russian, Japanese and German) perform better than the model from scratch, being Arabic the best model with a BLEU score of 23.33. In contrast, five languages performed worse than the model from scratch. This difference between languages is mainly caused by the grammatical similarity of the language to the Toki Pona.

However, there are more factors that can cause a difference in the score like the amount of data the model was trained with, the writing system, which might have been detrimental for the Chinese as it uses only a logographic system or many other peculiarities of the languages that are unknown to us.

With these results, we can also see that our model built from scratch performed very well taking into account that requires a total training much lower than the models which use transfer learning as their pretraining are very computing intensive, even though this work had already been done.

A surprising result is that the model pretrained with fewer examples, the Tok Pisin one, obtained a slightly higher score than the Spanish model, which is the one with more examples. This shows that, even with not that much data, if the language is similar enough, it can obtain a better score that a language which is

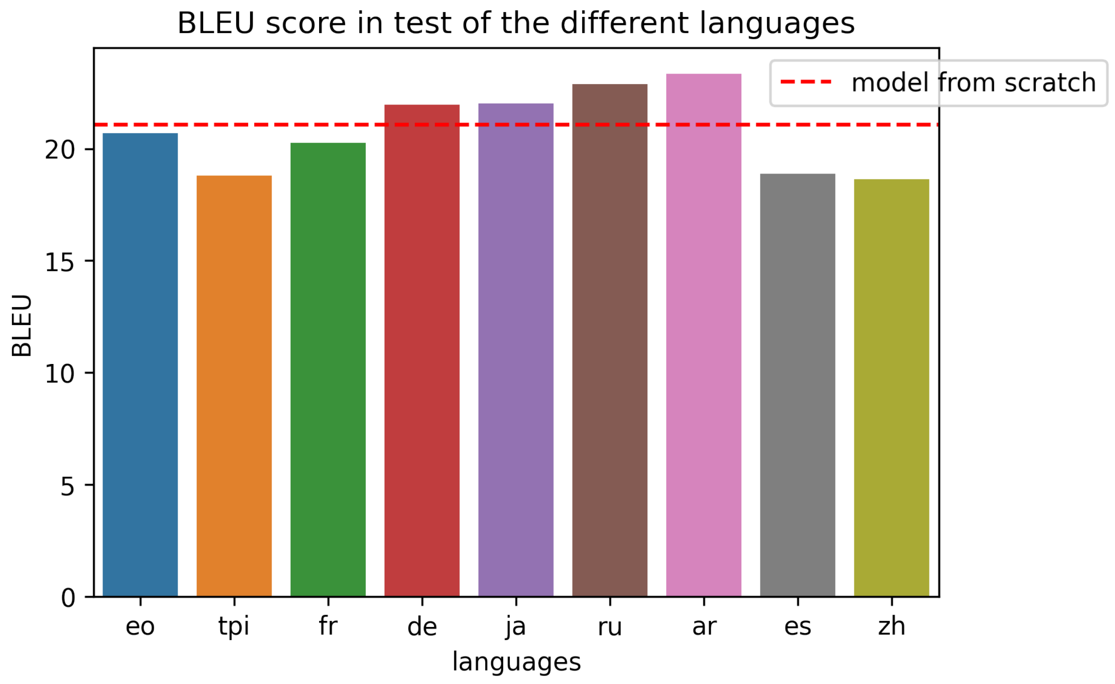

Figure 4.2: BLEU in the test set

very different but has been trained with many examples. This results also opens the door to many languages with low resources that have not been explored that might obtain a higher result than any of the languages tested in this work.

After seeing these results, we can conclude that even an artificial language like Toki Pona can benefit from transfer learning as there probably exists a language with enough similarity and resources to fine-tune from.

CHAPTER 5
Conclusions

After the development of this work, we reached the following conclusions:

- We built and validated a formal grammar for Toki Pona and used it to construct a sentence generator and a parser.
- An initial machine translator for Toki Pona trained from scratch has been developed and refined to obtain the best results.
- It was developed a Spanish to English translator using a public dataset.
- A study on transfer learning with very small models was performed leading to the conclusion that this technique benefits from being pretrained with a larger model.
- We benefited from large pretrained models from other languages to fine-tune them using transfer learning. Furthermore, we compared the performance of models of different languages and appreciated that some of them are beneficial for the task of translating Toki Pona and others are not.
- Our work, followed by some instructions for the use of the translator, have been released in open source so any person of the Toki Pona community can benefit from the translator and the rest of the tools developed in our investigation. The link that leads to the repository where the tools have been published is `https://github.com/pabagcha/toki-pona-tools`

5.1 Future work

In this project, we developed a functional Toki Pona translator. However, this translator could be improved by fine-tuning from models in other languages which might be more similar to Toki Pona. In addition, bigger models could also improve the performance of the translator but this type of models could not be explored due to our hardware limitations.

Furthermore, we only built a model to translate from Toki Pona to English so, benefiting from our research, future projects could create a model for translating from English to Toki Pona.

Finally, this research aims to establish a basis for the construction of linguistic tools for resource-poor languages. For this reason, we believe that our research could serve as a precursor for the future study of minority languages.

Acknowledgements

I would like to thank Damián López Rodríguez for directing this work and for believing in me almost two years ago when I sent him an email with some ideas that would result in this Bachelor's degree final project. Also, I have to thank Antonio Manuel Larriba Flor, experimental director of the project, for teaching me about Deep Learning and Machine Translation and for his patience, guiding me when I was blocked.

In addition, I would like to thank the Toki Pona community for solving some doubts that were necessary for this work and Sonja Lang for creating this language.

mi wile pona e kulupu pi toki pona tawa ni: sina pona e mi e pali ni. mi wile pona e jan Sonja Lang tawa ni: pali toki pona.

Bibliography

[1] Damian Podareanu, Valeriu Codreanu, Sandra Aigner, Caspar Leeuwen, and Volker Weinberg. Best practice guide - deep learning, 02 2019.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[3] Jaime Sevilla. Parameter counts in Machine Learning, July 2021.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, December 2017. arXiv: 1706.03762.

[5] S. Lang. *Toki Pona: The Language of Good*. Sonja Lang, 2014.

[6] V. Sartirani and S. Lang. *Toki Pona Dictionary*. Official Toki Pona. Amazon Digital Services LLC - KDP Print US, 2021.

[7] Itiroo Sakai. Syntax in universal translation. In *EARLYMT*, 1961.

[8] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 1958.

[9] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*.

[10] R.G.M Morris. D.o. hebb: The organization of behavior, wiley: New york; 1949. *Brain Research Bulletin*, 50(5):437, 1999.

[11] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *CoRR*, abs/2003.05991, 2020.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[15] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[16] Philip Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994.

[17] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, 2016. Association for Computational Linguistics.

[18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[19] M. A. K. Halliday. J. r. firth: Selected papers of j. r. firth, 1952–59. edited by f. r. palmer. (longmans' linguistics library.) x, 209 pp. london: Longmans, green and co. ltd., 1968. 30s. *Bulletin of the School of Oriental and African Studies*, 34(3), 1971.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, September 2013. Number: arXiv:1301.3781 arXiv:1301.3781 [cs].

[21] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Philadelphia, Pennsylvania, 2001. Association for Computational Linguistics.

[23] Iker García-Ferrero. Files in the repository, May 2022. original-date: 2019-01-11T19:56:57Z.

[24] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In John Hutchins, editor, *The Tenth Machine Translation Summit Proceedings of Conference*, pages 79–86. International Association for Machine Translation, 2005.

[25] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

[26] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[27] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[28] Jacob Eisenstein. Natural Language Processing. page 587.

[29] Nitin Indurkhya and Fred J Damerau. Natural Language Processing. *Machine Learning*, page 676.

[30] Noam Chomsky. On certain formal properties of grammars. *Information and Control*, 2(2):137–167, June 1959.

[31] Results of the 2021 Toki Pona census, June 2021.

APPENDIX A
Objetivos de Desarrollo Sostenible

Sustainable Development Goals (SDGs)	**High**	**Medium**	**Low**	**Not applicable**
SDG 1. **No Poverty.**				x
SDG 2. **Zero Hunger.**				x
SDG 3. **Good Health and Well-Being.**				x
SDG 4. **Quality Education.**	x			
SDG 5. **Gender Equality.**				x
SDG 6. **Clean Water and Sanitation.**				x
SDG 7. **Affordable and Clean Energy.**				x
SDG 8. **Decent Work and Economic Growth.**		x		
SDG 9. **Industry, Innovation, and Infrastructure.**	x			
SDG 10. **Reduced Inequalities.**	x			
SDG 11. **Sustainable Cities and Communities.**				x
SDG 12. **Responsible Consumption and Production.**				x
SDG 13. **Climate Action.**				x
SDG 14. **Life Below Water.**				x
SDG 15. **Life on Land.**				x
SDG 16. **Peace, Justice and Strong Institutions.**	x			
SDG 17. **Partnerships for the Goals.**				x

The aim of this work is to provide linguistic and translation tools to Toki Pona speakers and to make these technologies, which are normally only accessible to resource-rich languages, accessible and usable by this community of speakers. By trying to mitigate the gap between resource-rich and resource-poor languages, we are contributing to improving SDG 10: *Reducing Inequalities*.

According to a 2009 UNESCO report called UNESCO Interactive Atlas of the World's Languages in Danger, there are 2268 endangered languages and 230 languages that have become extinct since 1950 until the publication of the report. In our work, in addition to providing a translator for Toki Pona, we intend to lay a foundation for the treatment of minority language translation models that can be very beneficial for the preservation of these languages along with the traditions and culture of their speakers. In this sense, our work can contribute to SDG 16: *Peace, Justice and Strong Institutions* as it can help establish the linguistic institutions of these endangered communities.

Furthermore, the development of a translator in this language can help more people to learn the language since they will not need an expert Toki Pona speaker to translate the texts that the learner develops in the learning process. In addition, the use of a translator provides and enriches the grammatical knowledge necessary to speak, write and read sentences in a language. For all these reasons, we have considered that the work contributes to SDG 4: *Quality Education*.

In addition, this work promotes SDG 9: *Industry, Innovation and Infrastructure*. This is because it is a research that is framed in the field of artificial intelligence, Deep learning and NLP. These fields, in recent years, have become an essential part of scientific and technological development, creating tools that make our daily lives easier, such as the voice assistants we have in our mobile devices. These assistants make use of these technologies to make speech recognition, process requests and give the most satisfactory answers possible.

For this reason, technology companies recognize the importance of these technologies and have invested a great deal of money in them, making it a very profitable market in which the need for jobs for experts in these technologies has grown significantly. As our work is an advance in this direction, it could be considered to contribute slightly to SDG 8: *Decent Work and Economic Growth*.

However, many of the Sustainable Development Goals of the United Nations, such as those related to environmental or socioeconomic issues beyond those already mentioned, could not be addressed in this project for obvious reasons.

In conclusion, this type of work is of great importance in relation to several SDGs. The development of tools that can help communities with few linguistic resources can contribute to saving languages or even entire cultures. Helping these cultures is an essential task, not only for the communities to which they belong, but for humanity as a whole. Diversity shows us the incredible capacity of human beings to be creative and to be able to respond to a common problem, the need to communicate, in so many different ways. In my opinion, is in this capacity where the beauty of humanity lies.