UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dept. of Computer Systems and Computation

Evaluation of strategies for the adaptation of large neural models to the task of machine translation in constrained scenarios

Master's Thesis

Master's Degree in Artificial Intelligence, Pattern Recognition and Digital Imaging

AUTHOR: Iranzo Sánchez, Jorge

Tutor: Civera Saiz, Jorge

External cotutor: IRANZO SANCHEZ, JAVIER

ACADEMIC YEAR: 2022/2023

Resum

Històricament, la traducció automàtica (TA) ha sigut una de les àrees més actives dins de la intel·ligència artificial i, més precisament, dins del camp de l'aprenentatge automàtic. Gràcies a l'important progrés en l'entrenament de grans xarxes neuronals utilitzant grans col·leccions de dades que han aportat els principals proveïdors tecnològics, com Google, Meta, Microsoft, etc., la traducció automàtica multilingüe i els grans models de llenguatge s'han convertit en productes bàsics que aborden tasques àmplies que en alguns casos manquen d'especificitat. Encara que el rendiment general d'aquests models està fora de discussió, no és clar en quina mesura també aconsegueixen una precisió superior per a dominis específics amb accés limitat a grans infraestructures informàtiques. En aquest context, aquest treball avalua el rendiment de grans models quan s'adapten a tasques de TA amb factors limitants, com a especificitats de domini, pareixes d'idiomes involucrats i capacitat de còmput. Per a ser més precisos, aquest treball avalua l'aplicabilitat de models neuronals grans en comparació amb models basi sòlids en traduir de l'anglès a idiomes europeus dins del domini mèdic en el marc del projecte europeu INTERACT-EUROPE.

Paraules clau: Xarxes neuronals, Traducció automàtica, Adaptació de models grans

Resumen

Históricamente, la traducción automática (TA) ha sido una de las áreas más activas dentro de la inteligencia artificial y, más precisamente, dentro del campo del aprendizaje automático. Gracias al importante progreso en el entrenamiento de grandes redes neuronales utilizando grandes colecciones de datos que han aportado los principales proveedores tecnológicos, como Google, Meta, Microsoft, etc., la traducción automática multilingüe y los grandes modelos de lenguaje se han convertido en productos básicos que abordan tareas amplias que en algunos casos carecen de especificidad. Aunque el rendimiento general de estos modelos está fuera de discusión, no está claro en qué medida también logran una precisión superior para dominios específicos con acceso limitado a grandes infraestructuras informáticas. En este contexto, este trabajo evalúa el rendimiento de grandes modelos cuando se adaptan a tareas de TA con factores limitantes, como especificidades de dominio, pares de idiomas involucrados y capacidad de cómputo. Para ser más precisos, este trabajo evalúa la aplicabilidad de modelos neuronales grandes en comparación con modelos base sólidos al traducir del inglés a idiomas europeos dentro del dominio médico en el marco del proyecto europeo INTERACT-EUROPE.

Palabras clave: Redes neuronales, Traducción automática, Adaptación de modelos grandes

Abstract

Historically, machine translation (MT) has been one of the most active areas within artificial intelligence and more precisely, within the field of machine learning. Thanks to the significant progress in training large neural networks on massive collections of data brought to the table by major technological providers, such as Google, Meta, Microsoft, etc., multilingual MT and large language models have become staple products tackling broad tasks that in some cases lack specificity. Although the overall performance of these models is out-of-question, it is not clear to which degree they also achieve superior accuracy for specific domains with limited access to large computing infrastructures. In this context, this work evaluates the performance of large models when they are adapted to MT tasks with limiting factors, such as domain specificity's, language pairs involved and computing power. To be more precise, this work evaluates the applicability of large neural models in comparison with strong baselines when translating from English into European languages within the medical domain in the framework of the European project INTERACT-EUROPE.

Key words: Neural networks, Machine translation, Adaptation of large models

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

This work explores the current state of Machine Translation (MT), with a particular focus on the trend of using massively scaled-up models, often referred as "Large Language Models" (LLMs), whose popularity has risen in the recent years. Within this framework, we examine the potential of multilingual LLMs as translation models from English into other European languages in the context of a resource-constrained scenario. To achieve this, cost effective techniques are explored to adapt these models. These are compared to a more traditional bilingual MT baseline in order to assess the extent to which LLMs can truly deliver competitive results in MT.

This chapter contains the motivation and context of this work, as well as a brief introduction to key general concepts of Machine Learning that the reader needs to be familiar with in order to understand the rest of this work.

1.1 Motivation

With the rise of globalization during the past decades, an ever-increasing necessity of tools that facilitate cross-linguistic communication has appeared. In this context, the act of translation has been of great necessity in helping bridging the gaps of day-to-day communications.

One of the major breakthroughs that has lead to the wide availability of translation in the modern world is the appearance of automatic machine translation systems (MT) which are capable of achieving human-level translation accuracy. These cutting-edge systems, utilizing neural networks trained on extensive web-scraped datasets, offer a fast and dependable solution for translating text for the common citizen, institutions and companies alike.

In this regard, during recent years, major technological giants, such as Google, Meta or Microsoft, have been key in the development of the technology and at the forefront of implementing these systems in their commercial products. As such, through their economic power and general advances in hardware accelerators, there has been a increasing "gold rush" between these players to scale up these systems to never seen before heights at the promise of better performance and new capabilities.

This has led to the current landscape, where the list of publicly available neural networks capable of MT has been increasing at a surprisingly rapid pace. It is interesting to consider and study how well these large models, some of which have not even been directly trained for the MT task, compare to smaller and more affordable models in realistic scenarios, where the accuracy on specific domains and limited computing budgets may favor the latter.

1.2 Framework

This work has been done under the context of the **INTERACT-EUROPE** [1] project. This is an initiative by the European Union aiming to develop a European inter-specialty cancer training program involving all main oncology disciplines and professions. One of the main ideas of the project is to create an international curriculum for oncology experts which will be open to participants from various European countries. For this purpose, career specialists across Europe will have access to a series of training videos on the oncology domain.

Although the vehicular language of the project is English, not all cancer professionals master this language, and thus, the usage of ASR and MT adapted to the oncology domain has been one aspect that the project explores in order for the curriculum to reach all oncology professionals no matter their native language. In this context, the MLLP research group is responsible for the development of ASR and MT technology in the INTERACT-EUROPE project, with the work presented here corresponding to the latter part.

1.3 Objectives

The main objectives of this work are:

- To develop state-of-the-art MT systems for the usage in the **INTERACT-EUROPE** project.
- To adapt and evaluate the effectiveness of publicly available large models on MT.
- To explore how the latest advances and techniques in language modeling can be used in the creation of MT systems.

1.4 Document structure

Some parts of chapter 1 and 2 have been adapted from the bachelor degree dissertation of the author [2]. The document is structured into 6 chapters. Chapter 1 serves as a brief introduction to the field of Machine Learning. Chapter 2 introduces the field of modern MT, describing the Transformer architecture, as well as covering the usage of LLMs on MT and used evaluation metrics. Chapter 3 describes the datasets used for model training alongside how they were recollected and processed. Chapter 4 describes the training procedure and results on a series of baseline MT bilingual models in the context of the INTERACT-EUROPE project. Chapter 5 presents results on a series of adapted large multilingual models by taking into consideration the previous baselines and possible computing limitations. Lastly, Chapter 6 ends with a general overview of the conclusions and findings from this work, as well as a review on possible future work. In addition to the main chapters of this work, Appendix A provides additional figures related to training procedures and experimentation results.

1.5 Machine Learning

Within the vast field of Artificial Intelligence (AI), Machine Learning (ML) is in charge of the study and development of applications and systems capable of learning from data or past experiences, with the purpose of solving specific tasks and problems [3].

Fundamentally, the operation of these systems is governed by statistical models. The learning of the system consists in the search of some model that generalizes a set of data, i.e., that manages to predict and return the desired result when introducing new input data that has not been previously seen. Thus, the objective is that by means of some algorithm $\mathcal{A}$, through a set of training data, the optimum value of the parameters or weights that determine the output of the system is learned by a statistical model.

ML problems are usually categorized according to the domain of the system output. In classification problems, it is desired to predict the output y that takes values inside a set of $\mathcal{C}$ classes, such that $y \in \{1, \ldots, \mathcal{C}\}$. On the other hand, in regression problems one wishes to predict the output y that takes values within a non-numerable set A such as $\mathbb{R}$. In other words, in classification problems we output discrete values and in regression problems continuous values.

Depending on the availability of the correct output for given input data, ML algorithms are classified in different paradigms. In the context of this work, we primarily focus on supervised learning, which is typically used when dealing with problems in Natural Language Processing (NLP). In this paradigm, for each input data x, a y label is provided identifying the value that represents the sample. When training the model, a more appropriate decision can be made by varying the parameters $\boldsymbol{\theta}$ of the model and observing how much x differs or not from the y ground truth sample.

1.5.1. Neural Networks

Although today neural networks are the most common model in ML, the first notion of a neural network (NN) can be traced back to 1958, with the Perceptron [4]. This structure, together with the backpropagation algorithm [5] introduced in 1986, are the cornerstones of modern NN modeling.

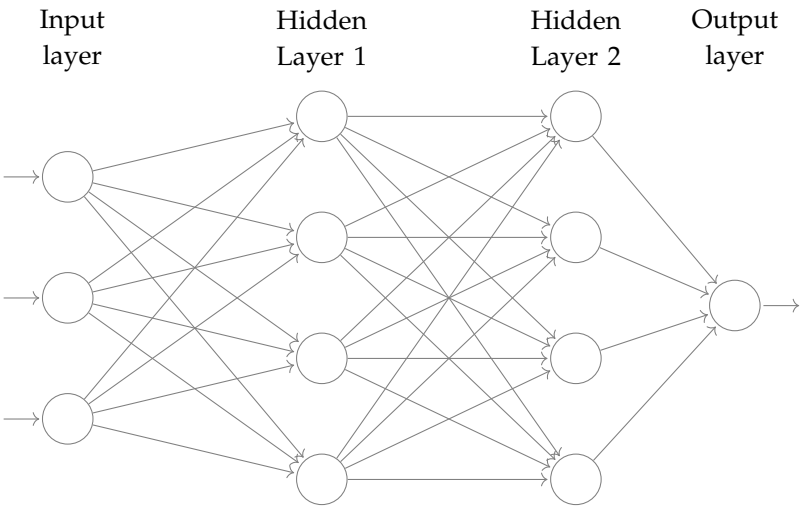

Figure 1.1: Example of a multilayer Perceptron with 3 neurons in the input layer, 2 hidden layers with 4 neurons each and an output layer with a single neuron.[1]

[0]Based on: https://tex.stackexchange.com/q/362238.

For a better understanding of the underlying operation of NN, the most basic network that implements these concepts, the multilayer Perceptron (MLP), is introduced, with an example of its structure displayed in Fig. 1.1. Formally, it is defined as a feed-forward neural network, in which a series of values flow through a graph from input to output, passing through a series of hidden layers, operating and modifying each output value of the previous layer. Each of these layers is made up of a number of nodes, commonly referred to as neurons, which are fully connected to all neurons in the next layer. The equations defining the values taken by the neurons in the hidden layer $\mathbf{h}^{(i)}$ are

$$\mathbf{h}^{(i)} = f^{(i)}(z^{(i)}) \tag{1.1}$$

$$z^{(i)} = \begin{cases} \boldsymbol{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} & if\ i = 1 \\ \boldsymbol{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)} & else \end{cases} \tag{1.2}$$

where:

- $\mathbf{x}$ is the initial vector of features supplied to the input layer.
- $\boldsymbol{W}^{(i)}$ is the matrix containing the weights of the incoming connections to the neurons of the hidden layer i.
- $f^{(i)}$ is the vector of activation functions of the layer. An activation function $f(\cdot)$ is usually defined by a nonlinear function that transforms the output of a neuron defined in turn by a function $z^{(i)}$. In this way, the model can learn nonlinear relationships. Examples of such activation functions are the sigmoid, the hyperbolic tangent or the ReLU [6].
- $\mathbf{b}^{(i)}$ is the vector of bias. These constants have the function of adding learnable parameters to each neuron so that the model is able to better learn the underlying function by giving it the possibility of adjusting and shifting the values corresponding to the output function of each neuron.

Over the years, different variations of neural networks have been developed based on the ideas put forward by the MLP. For example, Recurrent Neural Networks (RNN) [7], with its special variants such as LSTMs [8] or GRUs [9], allow the existence of cycles in their graph of connections to access information from previous time instants.

Globally, a neural network defines a discriminant function composed of other functions. It is desired that the optimal values of parameters $\boldsymbol{\theta}$ that define it, which in this case are the weights and bias described above, are learned by the model in order to optimize a cost or objective loss function $\mathcal{L}(\cdot)$.

In regression problems, this function is usually defined as the mean absolute error (MAE or $L1$)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid \tag{1.3}$$

or the mean square error (MSE o $L2$)

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1.4}$$

for the predicted values $\hat{y}_i$ and the real values y_i. On the other hand, for classification problems such as MT, the Cross-Entropy (CE) loss $H(p, q)$, which measures dissimilarity between two distributions, p and q, is a popular choice, and defined as

$$H(p, q) = \sum_{c} p(c)\log(q(c)) \tag{1.5}$$

The goal is therefore to minimize one of these functions. In this regard, the optimization comes from gradient descent, where for iteration i of the model, the parameters $\boldsymbol{\theta}$ are

$$\boldsymbol{\theta}_i = \theta_{i-1} - \rho \nabla_\theta \mathcal{L}(\theta) \tag{1.6}$$

with ∇_θ being the gradient of the loss function with respect to its parameters and ρ being a learning factor that regulates the update ratio of them. The premise of this method can be summarized as the computation of parameters that, knowing the steepness of the slope of the function and going in the opposite direction, brings us closer to a good minimum of the loss function.

The aforementioned backpropagation algorithm is in charge of the gradient calculation. The details of this algorithm are beyond the scope of this work, but it is recommended for readers which may not be familiar with it to read §6.5 of [10] for a better understanding of it.

It is important to note that the cost of computing the gradient is high, especially as the size of the datasets and the network increases, so modern neural networks typically make use of methods such as Stochastic Gradient Descent (SGD). In it, the gradient of the system is approximated by updating the underlying model in terms of aggregates of smaller data block referred as mini-batches, as opposed to calculating it with respect to the entire training dataset. Consequently, one may think that the resulting approximation of the global gradient of the model may be suboptimal, but the higher update frequency ends up resulting in a more robust convergence and computational efficiency, specially for the case of high-dimensional problems.

Another interesting fact to note is that, due to the way the gradient is calculated, if no preventive measures are taken sometimes the resulting values may turn out to be very small or big. In these cases, some weights may not be updated in a meaningful way by means of SGD and backpropagation and in the worst cases, may result into a situation where the network weights enter states where it can no longer be trained. These cases are often referred as situations where the gradient has vanished [11] [12] or exploded in value.

CHAPTER 2
Neural Machine Translation

This chapter introduces the main concepts, ideas and terminology of neural machine translation (NMT). The Seq2Seq paradigm is explored through an in-depth overview of the widely popular Transformer architecture. The second part of this chapter introduces the concept of Large Language Models (LLMs) and how they can be used for NMT, as well as highlighting the evaluation metrics that will be used with the rest of this work.

2.1 Machine Translation

We can define the goal of Machine Translation as that of constructing systems that are able to automatically translate texts from a natural language to another. Formally, given an input text $\mathbf{x} = x_1, x_2, \ldots, x_n$ made up of n distinct text tokens in a source language, we want to obtain the best possible translation $\hat{y}$ out of all $\mathbf{y} = y_1, y_2, \ldots, y_m$ possible translations in a given target language such that

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y \mid x). \tag{2.1}$$

Historically, the first viable MT systems were introduced during the late 1970s through the usage of rule-base techniques and methodologies that relied heavily on the knowledge of linguists and human experts [13]. Latter, during the early 1990s, Statistical Machine Translation (SMT) [14] emerged as an efficient data-driven alternative represented by the IBM models [15], [16] based on word-to-word alignments. This idea of SMT was latter improved and popularized through the 2000s by the introduction of phrase based models [17] and toolkits such as Moses [18], which worked by learning to translate significant "phrases" that were better in capturing translation subtleties.

During the early 2010s[1] these previous approaches were superseded by NMT that, with the progress on hardware accelerators and related software such as the CUDA Toolkit [20], proved to achieve better results and scalability. In its purest form, we can refer to NMT as the modelization of the MT process by a neural network.

For training NMT models, the CE loss (Eq. 1.5) can be used and rewritten as

$$-\sum_{t=1}^{n} \log p_\theta(y_t \mid y_{<t}, x) \tag{2.2}$$

where $y_{<t} = (y_0, y_1, \ldots, y_{t-1})$ indicates the partial target sequence, y_0 the special start token and θ the model parameters. As such, by optimizing the CE, we expect to optimize the output probability distribution over vocabulary $\mathcal{V}$ such that $p_\theta(\cdot|y_{<t}, x) \in \mathcal{R}^{|\mathcal{V}|}$ is the best approximation to the real MT mapping.

[1] There was work in NMT prior to the 2000s such as in [19].

2.2 Attention-based models

2.2.1. Seq2Seq and the attention mechanism

ML problems where input and output are sequences of variable length, such as in the case of MT, are usually denoted as *Sequence-2-Sequence*, or **Seq2Seq** tasks. In most of these types of problems, it is important for models to be able to compress, retain and recall relevant information for the task at hand across time and input dimension.

To deal with this type of problem, the use of models with a structure based on an encoder and a decoder was proposed in works such as [21]. The encoder, which historically has been implemented by a RNN, processes the input and produces a projection from a discrete $\mathcal{V}$ space of vocabulary tokens to a continuous one of dimension N that represents part of its features. This representation, referred to as the encoder's hidden states ($\mathbf{h}$), is processed by the decoder (usually another RNN), generating the desired output. Representations such as $\mathbf{h}$, which compress information about some input features, are known as *embeddings*.

However, a problem arises with this type of structure. The fixed, reduced dimensional nature of the hidden states $\mathbf{h}$ restricts the memory that the system has on the input. This not only affects the speed of inference, but gives bad results in sequences where the size is sufficiently large and there are long term dependencies in the sentence.

To solve this problem, in [22] the concept of attention was introduced. In this, we replace the vector of hidden states $\mathbf{h}$ by a context vector $\mathbf{c}$ that is dynamically computed for each iteration of the decoder such that:

$$\mathbf{c_i} = \sum_{j=1}^{N} \alpha_{ij} h_j \tag{2.3}$$

where the weights $\alpha_{i,j}$ are calculated for each hidden state h_j using the *softmax* function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{N} \exp(e_{ik})} \tag{2.4}$$

and in Seq2Seq problems

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.5}$$

where s_{i-1} is the hidden state of the system after the last decoder iteration and $a(\cdot)$ a scoring function that returns the degree of attention or energy e_{ij}, typically implemented by a feed-forward layer.

An alternative way to view the attention mechanism, as defined in [23], is as a lookup over a dictionary. In this view, we compare a query value $\boldsymbol{Q}$ to a set of key-value pairs $(\boldsymbol{K}, \boldsymbol{V})$, returning a weighted sum of the values, where the weight assigned to each value is calculated by a function that measures the compatibility of the query with the corresponding key. For example, in the above case the compatibility function would be Eq. 2.4.

2.2.2. Transformers

The Transformer architecture, originally introduced in [23], emerged as a substitute to the use of RNNs using the attention mechanism introduced in Sec. 2.2.1. The general encoder-decoder structure proposed in [23] can be seen in Fig. 2.1.

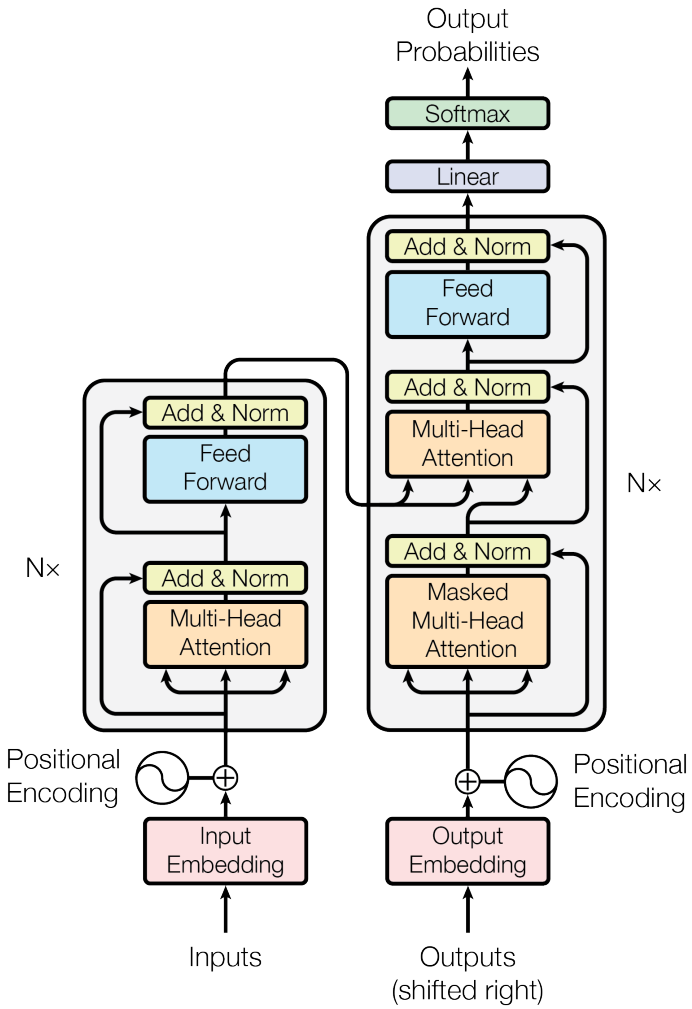

Figure 2.1: Original Transformer architecture (Post-LN) as seen in [23].

The main components and ideas of this Transformer are the following:

- **Scoring function:** Taking the definition of attention as the $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ dictionary lookup presented in Sec. 2.2.1, we define the $a(\cdot)$ as the dot product and scale results by the inverse factor of the $\boldsymbol{K}$ dimension size, d_k, passing the result through a *softmax* such that

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{2.6}$$

- **Multi-Head Attention:** Instead of computing a single attention function for each query $\boldsymbol{Q}$, multiple attention functions, or *heads*, are computed with projections for each set of queries, key and value $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$, which are then concatenated and re-projected into a common space $\boldsymbol{O}$

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, \dots, head_h)\boldsymbol{W}^{\boldsymbol{O}} \tag{2.7}$$

and

$$head_i = Attention(\boldsymbol{Q}\boldsymbol{W}_i^{\boldsymbol{Q}}, \boldsymbol{K}\boldsymbol{W}_i^{\boldsymbol{K}}, \boldsymbol{V}\boldsymbol{W}_i^{\boldsymbol{V}}) \tag{2.8}$$

where W_i^Q, W_i^K, W_i^V and W^O are the corresponding learnable matrices.

This procedure allows the system to adapt and identify similarities and dependencies in various dimensional ranges. In addition, by splitting the attention computation, the number of sequential operations is reduced, allowing a higher degree of parallelization of the model, and therefore, higher training speeds.

Inside the Transformer, depending on the nature of the input vectors, we can distinguish two distinct attention blocks:

  - **Self-attention**: Where Q, K, V correspond to the output of the previous layer of the encoder or decoder.
  - **Cross-attention**: When in the decoder, Q is taken from the decoder previous layer, and K, V are extracted from the output of the last encoder layer. This way, each decoder position can attend to all positions of the input stream simultaneously.

- **Causal masking**: To prevent the usage of future information, the values of the initial self-attention block on the decoder are masked to $-\infty$ before the *softmax* so that predictions only attend to tokens at previous positions.

- **Positional Encodings:** In RNNs the order of the input tokens can be easily inferred, since it is implicitly defined with the sequential analysis that there are token to token. However, with the Transformer structure this information disappears and some other mechanism has to be implemented to extract the relative or absolute position of the tokens in the input sequence. For this, the original authors propose a representation that is summed to the outputs of the initial word embedding layer, with the position of each token implicitly represented in a vector whose values are calculated based on a sine function that is described as

$$PE_{pos,2i} = sin\left(\frac{pos}{G^{2i/d_{model}}}\right) \tag{2.9}$$

where pos_i corresponds to the input position with respect dimension i, and the constant G, which in the original paper takes the value of 10.000, restricts the function into a specific range. More precisely, this function represents a geometrical progression in the range of $[2pi, 2piG]$, which authors hypothesize facilitates the model in the process of learning to attend to relative positions. For a more detailed view behind the reasoning of this mechanism, we refer the reader to [24].

- **Normalization and skip connections:** To better control the gradient flow in the network and improve training stability, the Transformers blocks include skip connections [25] and layer normalization [26]. For the latter, in the original Transformer paper this component is placed after each residual block (Post-LN). However, contrary to this, in the original Tensor2Tensor implementation [27], the layer norm was placed in the residual before the attention and feed-forward layers (Pre-LN). This second configuration has been found to generally be more stable, specially when scaling up the number of parameters of Transformers [28].

2.2.3. Scaling the Transformer Architecture

The following is a list of different trends and best practices popular in Transformer training and inference which improves it in terms of efficiency, generalization and adaptation capabilities [29]:

- **Multi-GPU training**: With the increasing complexity and size of Transformer models, training on a single GPU can become time-consuming and computationally intensive. For effective scaling of Transformer models, proper usage of multiple GPU training is a must. By harnessing the power of multiple GPUs, the training of a model can be effectively parallelized and scaled up along different nodes, and at the same time, allow bigger batch sizes, which have been proved to be important to obtain good results in Transformers [30].

  Although hard to implement by their own, fortunately the majority of deep learning toolkits, such as PyTorch [31], already implement ready-to-use wrappers of distributed training paradigms. For smaller sized models, **Distributed Data Parallel (DDP)** [32] is a popular choice. In DDP, a process is assigned to each computing device with its own local model replica and optimizer, receiving at the forward step a portion of the input chunked along the batch dimension. Then, during the backward pass, the gradients are synchronized making sure that the all replicated model end up at the same updated final state.

  Although effective, two major problems appear when using DDP: One, is that the model needs to fit on all GPUs used in the training procedure, and two, is that at large scale training, model replication derives in a major overhead of memory usage. A popular alternative to DDP is **Fully Sharded Data Parallel (FDSP)** [33], which is inspired by stage 3[2] of the ZeRO optimizer (ZeRO) [34]. In essence, in FDSP the optimizer states, gradients and model parameters are sharded into units along all devices, which are later communicated and recovered on-demand before computations are made through the usage of distributed computing primitives such as gather and scatter. Additionally, CPU offloading of optimizer states and gradients is habitual in FDSP.

- **Half-precision training and quantization**: Without taking any special measure, the choice of numeric precision of model can easily become a major computing bottleneck, specially as parameter count increases in large scale training. Historically, neural networks have been trained in `float32` precision, but current hardware accelerators support faster training with lower precision, without any performance degradation. In this regard, `float16` training, as presented in [35], is widely used in the training of Transformers, and other floating point types such as `bfloat16` [36] and `float8` [37] remain as popular choices. In this regard, `bfloat16` offers similar results to `float16`, but it has been observed to have better stability during training, while `float8` is considerably faster but more unstable than the 16-bit variants.[3]

  Moreover, integer types have also been widely explored for model quantization, particularly in regards to model inference, such as in the case of `int8` [38], which remains a popular choice for its balance between stability and speed. Additionally, there also has been recent works that leverage `int4` formats [39] for LLM adaptation, and there has been even cases where with some architectural changes to the Transformer, 1-bit model training is possible [40].

- **Gradient Accumulation**: As previously mentioned, Transformers benefit greatly through the usage of big batch sizes, but it is quite common in practice to find that the necessary batch size for a given model size does not fit into device memory. A popular technique to alleviate this and simulate larger batches is gradient accumulation [41]. In it, instead of updating the network weights on every batch, gradient values are accumulated for each batch and the backward pass is delayed by K steps.

[2] Accordingly, stage 1 of ZeRO only shards optimizer states, while stage 2 also shards gradients.
[3] `float8` was introduced for NVIDIA Hopper hardware accelerators in late 2022.

- **Data Augmentation**: Compared to other traditional ML models, Transformers are quite data-hungry and require to be trained in the ranges of billions or even trillions of tokens. The curation of large-scale, high-quality datasets, is then key in order to obtain good performance of Transformer models. However, more often than not, data recollection can be challenging for many domains, with situations where data may be limited, expensive to collect, or subject to privacy concerns. Moreover, datasets may lack diversity, representing only a narrow range of instances and failing to capture the variability present in real-world scenarios. For these reasons, the creation of synthetic data, or data-augmentation, which seek mitigate these problems, is a popular topic research among NLP practitioners that use Transformers.

  In the context of MT, a popular data-augmentation strategy is the creation of additional parallel corpora from monolingual data. In parallel MT corpora we can distinguish phrases from the direction that they were translated from, that is, if they are *source-original* or *target-original*. Following this taxonomy, we can classify MT data-augmentation techniques if they create data from source-side or target-side phrases, which are named accordingly as *forward-translations* (FT) [42] and *back-translations* (BT) [43], with the latter technique having been found to be generally more effective than the former [44].

  For BT, the synthetic source-side is typically obtained from a reverse translation model, trained with the corresponding $src \rightarrow tgt$ data to be used on the final model. Additionally, the usage of "tags" [45] to distinguish original-source-side phrases in BT has been found to decrease quality degradation that can be introduced by the technique such as the case of *translationese* artifacts [46], [47].

Apart from the previous points, there exist in the literature many modifications to the Transformer that improve its base performance which we have omitted for the sake of brevity. Some examples of the most popular modifications are the usage of alternative positional embedding formulations [48], [49] and activation functions [50] or faster decoding methods [51], [52] and attention calculations [53].

2.3 Large Language Models

Although originally a model specialized for NMT, the Transformer architecture explained in Sec. 2.2.2, through its derivatives, is currently one, if not the most popular architecture for text-based ML tasks. Apart from this, the Transformer and its ideas have seen a wide usage in other realms such as Computer Vision [54], Automatic Speech Recognition [55] or Text-to-Speech [56], [57], obtaining state-of-the-art results in these tasks. The main key to the success of the Transformer architecture can be mostly attributed to four main components:

- The parallel processing capabilities and speed.
- The ability to handle long-range dependencies when compared to older architectures.
- The adaptability of the model to any type of input sequences.
- The positive response to scalability regarding size and data for obtaining better results.

For these reasons, ever since its popularization, the Transformer has successfully served as a standard solution for many problems in ML. With the promise of Transformers to serve as a "jack-of-all-trades", each passing year researchers and companies

alike have been scaling up these models to sizes that a few yeas ago would be considered unheard of. As an example, compare the size of the original Transformer 300-million parameter variant, which was regarded at the time of release as a relatively big model, to the 175-billion parameters count of GPT-3 [58] or 540-billion of PALM [59].

These pre-trained large Transformers, along other contemporary models based in language modeling tasks [60], [61] with a similar parameter size and computing budgets, are currently widely referred under the umbrella term of **Large Language Models**, or **LLMs** [62], and have seen a major increase of interest both in industry and mainstream discourse of ML.

A popular taxonomy used to identify LLMs is the division between three large families of Transformer models depending on whether they make exclusive use of the Transformer encoder, decoder or both, such as in the original paper. In this regard we can identify, BERT [63], GPT [64] and T5 [65] as the first works which popularized each described variant for general purpose usage. It is important to note that, for the most part, decoder-only models have remained as the most popular choice in recent years, and as such, by classifying a model as a "LLM" it is often implied that it is a decoder-only model.

2.3.1. In-Context Learning

One of the key concepts that is fundamental to the effectiveness and versatility of LLMs is that of In-Context Learning (ICL) [66], which was originally introduced in [58]. In the latter work, ICL is referred as the capacity of a given pre-trained language model or LLM to be conditioned on a natural language instruction and/or a few demonstrations of a given task, and subsequently, solving it by treating it as language modeling task. That is, by simply predicting the most probable tokens that follows the conditioning input, a LLM can resolve complex Natural Language Understating and Generations tasks (NLU, NLG) that were not seen during training given enough input information.

A common way to refer to these conditioning inputs on the literature is "prompts", with each example of the task that is passed to the input counting as a "shot". Accordingly, when no example is present in the input a "zero-shot" setting occurs, which contrasts with the *k*-shot or "few-shot" setting in the presence of them. Depending on the model, the quality of prompt and shots can greatly impact downstream task performance. As such, when leveraging ICL, formatting of the prompts with techniques such as Chain-of-Thought (COT) [67] or adequate selection methods of them is crucial. Additionally model parameter size has been observed to greatly influence the capability and effectiveness of ICL.[4]

While the exact workings of ICL in LLMs are not yet fully understood, there have been recent works exploring this topic that seems to indicate that ICL in LLMs may be attributed by the ability of large enough models to learn adaptable functions that are implicitly "fine-tuned" through gradient updates in the model's forward passes [71], [72], [73].

[4] In the LLM literature it has been observed that, at certain size thresholds, some models enter "phase transitions" where "emergent capabilities" [68] for certain tasks may appear unpredictably and in a sharp manner that were not previously seen in smaller models. However, this view has been recently criticized in [69], [70], which respectively attribute these results to the innate capabilities of ICL and the misuse of evaluation metrics.

2.3.2. Applying LLMs to MT

Small bilingual encoder-decoder models have remained, for the most part, as the go-to architecture when training NMT models. Nevertheless, recent work showed that decoder-only language models can achieve similar, if not better results for MT, when simply concatenating the source phrase in the input [74], [75] [76]. This discovery, along the ICL capabilities of LLMs, has lead to an explosion of works during the last year exploring the capabilities in MT of multilingual decoder-only LLMs which have not been explicitly been trained in the task. As examples, the following works explore MT for different type of LLMs [77], OpenAI's GPTs [78] [79] [80], BLOOM [81], XLGM [82], GLM [83], and PALM [84]. From the previous mentioned works, we can extract the following points regarding LLMs usage in MT:

- LLMs translations can produce higher quality metrics that tend to be less literal [85], as well as having different types of translation errors compared to regular NMT models [86]. This has been pointed out in works such as [84], [86] to have the effect of discrepancies between traditional count based and neural metrics on resulting LLMs translations, with the former seemingly being less accurate than the latter at reflecting human preferences [87].
- Precise prompts and on-topic shots can greatly help in boosting translation quality and resolving uncertainties. Prompt adaptation can be easily used to naively enforce specific tones or glossary for translation [88].
- Part of the MT capabilities of LLMs can be partially explained due to bitext contamination present in the training data [89]. This raises questions at how beneficial is training LLMs with general data if MT is the downstream objective.
- "Bigger is better", in the sense that model quality for LLMs in MT tend to follow general LLMS scaling laws [90], as well as observed bilingual [91] and multilingual [92] MT scaling laws.

2.3.3. Parameter Efficient Fine-Tuning

After the training process of a model, it is quite often common to further train or "fine-tune" it[5] to improve performance on a downstream task or domain. Some general approaches to fine-tuning rely on modifying all model parameters or partially freeze some layers and blocking gradient updates on them. However, applied trivially, the former can become quite expensive, specially for larger models, and prone to suffer from *catastrophic forgetting*, while in the latter, layer selection can be cumbersome, with results that are often suboptimal in relation to the ones obtained in full parameter fine-tuning.

For these reasons, in the wake of LLMs, the ML community has been in the search of Parameter Efficient Fine-Tuning (PEFT) [93] methods that take into account possible computing resources limitations, updating only the necessary amount of model parameters for the adaptation of it to a new task. Out of these methods, LoRA [94], as presented in Fig. 2.2, is one of the most popular PEFT architectures. It is mostly based on the observation that large enough models tend to have a low intrinsic dimension, that is, that within a certain level of approximation error the learned function by the model can be effectively represented in a small dimensional subspace [95], [96]. Following this, LoRA introduces the necessary updates to adapt to a new task as learnable small low-rank matrix decompositions of selected W dense matrices of the original model, which is in turn freezed

[5] By fine-tuning, we refer in this case to parameter fine-tuning, as opposed to hyperparameter exploration and optimization.

completely during the fine-tuning procedure. In the case of Transformers, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ and $\boldsymbol{O}$ attention matrices are usually selected to be updated by the W matrices. More formally, for each weight matrix W_0 update ∇W we constrained them such that for $\nabla W = BA$

$$h = W_0 + \frac{\alpha}{r} BA \tag{2.10}$$

where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ are the learnable decompositions with rank $r \ll \min(d, k)$ and scaling factor α. In the original implementation, A and B are respectively initialized to a random Gaussian distribution and zero. After finishing training, ∇W adapter weights can be merged with the base model for zero additional inference latency, which contrasts with other adapter methods that with the introduction of additional parameters make the resulting model slower and heavier in size.

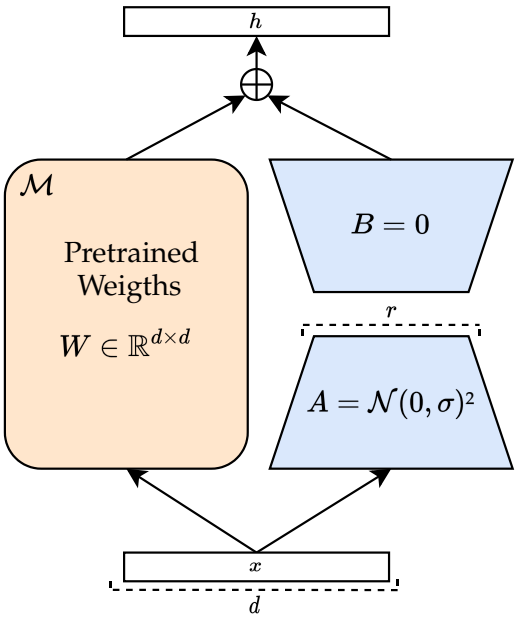

Figure 2.2: LoRA architecture. Adapted from [94].

2.4 Evaluation Metrics for Machine Translation

Ideally, MT systems are to be evaluated manually by human experts, which ultimately are the final judges which can assess the quality of a given translation. However, these evaluations are costly and cannot be easily scaled, so this has led the MT community to the creation of fast automatic evaluation metrics that try to correlate as close as possible to human judgment, This has allowed the scientific community to easily evaluate and compare performance of several models during the experimentation phases of their work and against results of other research papers.

Historically, the definition of automatic evaluation metrics in MT has been a hot topic among researchers. Although there is a clear degree of underlying "objective quality" between a translation and its original text, there also lies a clear degree of subjectivity and fuzziness in translation that makes it difficult to solidify one quality measuring metric as a definitive one. Compare this, for instance, to the case of ASR, where the scientific community has converged into a *de facto* usage of Word Error Rate (WER) and Character Error Rate (CER) as quality metrics.

The underlying evaluation problem of MT can be easily traced back to the one-to-many nature of the task, where more than one target translation can be correct, combined

with the amount of ambiguities that can easily appear with the usage of different registers or dialects inside a language. Nevertheless, even with these problems, through the years the usage of automatic quality measuring metrics has been solidified as the standard way of evaluating different MT models.

Count based metrics

In terms of traditional automatic metrics, the most popular metric used in MT is the *Bilingual Evaluation Understudy*, or BLEU [97]. This metric is based on an modified average n-gram based precision p_n, such that it clips precision with the assumption that n-gram should appear at maximum the same amount of times in both the reference r and a candidate c translation. We calculate

$$AveragePrecision(N) = \frac{1}{N}\sum_{n=1}^{N} \log p_n \tag{2.11}$$

with the precision weighted by a brevity penalty between reference and candidate lengths

$$BrevityPenalty = \begin{cases} 1 & if\ c_l > r_l \\ e^{(1-\frac{r_l}{c_l})} & if\ c_l \leq r_l \end{cases} \tag{2.12}$$

such that, typically with N=4, we have

$$BLEU = BrevityPenalty * AveragePrecision(4) \tag{2.13}$$

giving a value between 0 and 1 that is typically multiplied by 100 for better readability, with higher values denoting better translations.

Apart from BLEU, other popular count based metrics are the n-gram based chrF [98], [99] or the edit-distance based Translation Error Rate (TER) [100]. Alongside BLEU, these metrics can take different parameters, and as such, inconsistencies can appear that distort model evaluation, specially if text normalization and tokenization between hypotheses and references is not preserved. For this reason, it is standard to calculate these three metrics through usage of the SacreBLEU library [101] for the sake of consistency. Apart from this, it is heavily recommended when reporting results to attach the output signature of this tool which describes metrics parameters for better result reproducibility.

Neural based metrics

These metrics are mostly based on the usage of the embedding representation of pre-trained neural encoders such as BERT for MT evaluation. Overall, these metrics have been found to generally better correlate with human preference compared to traditional count based metrics [102], [103]. In [104], the authors identify two mains approaches in the literature:

- **Embedding-distance metrics**: These metrics substitute the word/n-gram matching of classical metrics by leveraging the fuzziness introduced by distance similarities scores between embedding representation from pre-trained models for capturing semantic similarity without making modifications to the representation: BERTScore [105].
- **More recent fine-tuned metrics**: Which modify the underlying embedding representation to try to better adapt the models to the MT quality assessment task,

adding additional training objectives to make the resulting learned metrics more sensitive to specific errors or *hallucinations* [106] in translations:[6] BLEURT [108], [109] and the COMET family [110], [104].

COMET-22, which will be used to compare models in future chapters, is one of the current neural based metric which has seen wide adoption by the community. This metric makes use of a pre-trained, multi-lingual model XLM-RoBERTa [111], which has been modified and trained on high quality translation pairs in order to use it as a translation quality score regressor. This metric works as follows: First, source, hypothesis, and reference sentences are encoded into multiple word embeddings from intermediate layers of the model, which are then passed through a trainable layer-wise attention mechanism, concatenated and average pooled, resulting in sentence-level embeddings vectors $\boldsymbol{h}$, $\boldsymbol{s}$ and $\boldsymbol{r}$. These are then transformed into multiple representations $\boldsymbol{h} \odot \boldsymbol{s}$, $\boldsymbol{h} \odot \boldsymbol{r}$, $|\boldsymbol{h} - \boldsymbol{s}|$, and $|\boldsymbol{h} - \boldsymbol{r}|$, which along $\boldsymbol{h}$ and $\boldsymbol{r}$ are concatenated and feed into a trainable forward regressor that outputs scores between 0 and 1, with the minimum indicating random input and the maximum, a perfect translation. The resulting score are normally reported by further scaling them to 0 and 100.

Other remarks on MT metrics

Currently, the majority of MT metrics are *reference-based*. These compare output translations of a model respect a ground-truth translation obtained from human experts. Although more expensive than *reference-free*[7] solutions such as Round-Trip Translation [113], these tend to be preferred by the MT scientific community since they have historically given more precise estimates.

Another important fact to have in mind is that all previously named metrics make evaluation at an individual sentence level, and require a previous segmentation before evaluation. In more specific tasks like Speech Translation [114] or Document Level MT [115], [116], [117], the previous metrics tend to be suboptimal, and more specific metrics are used.

[6] As an example, the Robustness MT evaluation framework *SMAUG* [107], identifies deviations in named entities, numbers or meaning, or the insertion and removal of content, as typical errors in MT systems.

[7] In the literature, *reference-free* metrics are also typically referred as "Quality Estimation" (QE) [112].

CHAPTER 3
Datasets

This chapter introduces the dataset compilation process for training the baseline models of the INTERACT-EUROPE project. Furthermore, the used data-processing pipeline is discussed alongside the tokenization scheme that will be used for the models of the following chapters.

3.1 Datasets

3.1.1. Evaluation sets

The European School of Oncology[1] (ESO) provided a series of video conferences with English transcriptions from which a subset was sampled to select development and test sets for model evaluation. More precisely, a series of 10 videos were split into two subgroups for a total of 3.5h and 3.8h of speech respectively, which were given to a professional translation agency[2] resulting in a series of non-aligned translations from English to French, Spanish, German and Slovene. From these, a first pass clean-up to the text was made with a curated list of regular expressions. Following this, sentence-level bitext was obtained for each language direction through the extraction of phrases of the evaluation sets with the Moses toolkit script `split-sentences.perl`, which were further processed with the neural based alignment tool Vecalign [118]. Finally, a second manual clean-up was made to correct alignment errors and mistakes which were not detected on the initial clean-up. The total number of sentences for each language pair evaluation sets is presented in Table 3.1.

Table 3.1: Total number of sentence-level bitext for INTERACT-EUROPE evaluation sets.

$en \rightarrow$	Dev	Test
fr	1445	1407
es	1450	1405
de	1424	1399
sl	1458	1407

[1] https://www.eso.net/

[2] All translators were asked to follow their usual translation workflow except for two points: Translations must be obtained from scratch without the usage of any MT tooling, and literal translations were to be preferred.

3.1.2. Training sets

For the training of baseline models, parallel corpora from all explored language directions was mainly extracted from the OPUS [119] platform.[3] These corpora are made out of a mix of filtered web-crawled data such as Paracrawl [120] and high quality texts derived from sources such as books or parliamentary debates, as is the case of UNPC [121]. For the case of $en \rightarrow de$, the WMT22 main shared task data recipe was followed.[4] In addition to this, for $en \rightarrow es$, de, fr additional data from Medline abstracts was added based on the list provided by the WMT22 Biomedical Translation Task [122] [5], which were processed into bitext by the `split-sentences.perl` and Vecalign procedure described in Sec. 3.1.1.

For the case of $en \rightarrow sl$, due to the data scarcity compared to other language pairs, back-translations were considered with additional Slovene data that was scrapped from oncology journals and books sources presented by the Institute of Oncology of Ljubljana [6] along the usage of the SiParl [123] corpora. Apart from these, the incorporation of noisier bitext was considered by testing the effect of the CCMatrix [124] corpus on the resulting models. The previously selected data was cleaned by discarding non-Slovene phrases chosen by the language identification model `lid.176.bin` from fastText [125] combined with the fixing and elimination of low scoring sentences by the Bifixor and Bicleaner tools [126], [127].

A overview of bitext corpora selected for each language pair is reported in Tables 3.2, 3.3, 3.4 and 3.5 respectively, as well as the extra monolingual data for the $en \rightarrow sl$ direction in Table 3.6. Overall, total data amount rounds up to 340.9 million sentences for $en \rightarrow es$, followed by 309.0M for $en \rightarrow fr$ and 289.0M for $en \rightarrow de$, with $en \rightarrow sl$ around the 74.8M mark when counting the extra bilingual and monolingual data.

Table 3.2: General domain corpora for language pair $en \rightarrow fr$, where K=10^3, M=10^6 and G=10^9.

Corpus	Bitext	Words	
		English	French
ParaCrawl [120]	216.6 M	3.7 G	4.1 G
UNPC [128]	30.3 M	658.4 M	816.4 M
Giga Fr-En [129]	22.5 M	575.8 M	672.2 M
EUBookshop [130]	10.8 M	224.6 M	244.5 M
CCAligned [131]	15.6 M	156.7 M	171.1 M
DGT-TM [132]	4.9 M	86.3 M	95.4 M
WikiMatrix [133]	2.7 M	57.8 M	63.1 M
WikiMedia [129]	1.0 M	24.1 M	25.8 M
Europarl [121]	1.2 M	28.6 M	29.9 M
News Commentary [129]	3.2 M	70.7 M	76.6 M
CommonCrawl[7]	0.1 M	4.1 M	4.7 M
Medline-WMT22	110.6 K	2.4 M	3.0 M
Europarl-ST [134]	96.5 K	2.3 M	2.6 M
Total	309.0 M	5.6 G	6.3 G

[3] https://opus.nlpl.eu/

[4] https://www.statmt.org/wmt22/mtdata/index.html

[5] https://github.com/biomedical-translation-corpora/corpora

[6] https://www.onko-i.si/eng/sectors/research_and_education/medical_and_other_scientific_publication

[7] https://commoncrawl.org

[7] https://europat.net/

Table 3.3: General domain corpora for language pair *en* → *es*, where K=10^3, M=10^6 and G=10^9.

Corpus	Bitext	Words	
		English	Spanish
ParaCrawl [120]	269.4 M	5.0 G	5.4 G
EuroPat [8]	51.4 M	1.7 G	1.8 G
MultiUN [135]	11.4 M	29.7 M	330.2 M
DGT-TM [132]	5.2M	113.5 M	121.7 M
Wikipedia [136]	1.8 M	41.2 M	44.9 M
Europarl [121]	1.1 M	28.8 M	27.6 M
Medline-WMT22	153.8 K	3.6 M	4.2 M
Total	340.9 M	6.9 G	7.7 G

Table 3.4: General domain corpora for language pair *en* → *de*, where K=10^3, M=10^6 and G=10^9.

Corpus	Bitext	Words	
		English	German
ParaCrawl [120]	278.3 M	4.2 G	4.0 G
TildeMODEL [137]	5.1 M	131.4 M	108.8 M
CommonCrawl-WMT13 [138]	2.3 M	58.8 M	54.5 M
WikiTitles-WMT19 [139]	1.4 M	44.0 M	35.9 M
Europarl [121]	1.1 M	27.9 M	25.8 M
News-Commentary [140]	388.4 K	9.2 M	9.1 M
Europarl-ST [134]	105.3 K	2.3 M	2.2 M
Medline-WMT22	95.5 K	2.0 M	2.2 M
Total	289.0 M	4.4 G	4.1 G

Table 3.5: General domain corpora for language pair *en* → *sl*, where K=10^3, M=10^6 and G=10^9.

Corpus	Bitext	Words	
		English	Slovene
OpenSubtitles [141]	19.6 M	129.0 M	98.0 M
ParaCrawl [120]	9.5 M	151.6 M	137.8 M
DGT-TM [132]	5.1 M	86.5 M	76.0 M
TildeMODEL [142]	2.0 M	42.2 M	38.2 M
EMEA [9]	1.0 M	11.7 M	11.6 M
Europarl [121]	633.4 K	15.0 M	12.5 M
EUbookshop [10]	405.6 K	8.8 M	7.9 M
EuTV [11]	181.1 K	1.7 M	1.4 M
Wikipedia [136]	140.1 K	2.2 M	2.9 M
JRC-Acquis[12]	53.3 K	879.1 K	724.5 K
Ted2020 [143]	44.3 K	733.4 K	443.3 K
WikiMedia [129]	31.7 K	806.6 K	689.1 K
WIT3 [144]	17.1 K	285.4 K	235.5 K
Total	38.8 M	452.6 M	387.1 M
Extra: CCMatrix [124]	27.4 M	364.9 M	216.4 M

Table 3.6: Monolingual corpora for language pair *en* → *sl*, where K=10^3, M=10^6 and G=10^9.

Corpus	Monotext	Words
SiParl [123]	8.7 M	192.5 M
Oncology-Institute	36.1 K	859.1 K

3.1.3. Data processing pipeline

All data is pre-tokenized using the Moses cleaned with Moses `clean-corpus` script to discard sentences with more than 250 tokens. After that, a Truecasing [145] model is used to reduce overall vocabulary size. Truecasing can be seen as a normalization technique where a model tries to transform text into its most appropriate form in upper or lower case by collecting different statistics and building a prediction model out of these. In Moses, the Truecasing model only changes the words at the beginning of a sentence to their most common form, as well as any words in which their current form is unknown.

All data is then tokenized with a trained subword vocabulary from the collected data, which offer an economic way of simulating large vocabularies and solve OOV problems. As a way to illustrate the benefits of subword tokenization, consider the following example. Let $\mathcal{V}$ be a word-level Spanish vocabulary such that $\mathcal{V} = [\texttt{traducción}, \texttt{conducir}]$. A model trained with $\mathcal{V}$ would only be able to identify these words. However, a model trained with a vocabulary made out of subword $\mathcal{V} = [\texttt{ción}, \texttt{tra}, \texttt{duc}, \texttt{con}, \texttt{ir}, \texttt{duc}]$ could additionally identify words such as `ir`, `conducción` or `contra`.

For subword-based tokenization, in this work we use the SentencePiece library [146]. SentencePiece provides a robust and fast C++ implementation of a subword tokenizer with various partitioning algorithms such as Byte-Pair Encoding [147], [148] and unigram-based language models [149]. This library also offers other benefits such as automatic NFKC UTF-8 text normalization, possible regularization of subwords and language agnostic representations through special treatment of whitespaces.[13]

For this work, BPE was chosen as the partitioning algorithm. Roughly speaking, BPE fragments the text at character level and learns text joins (*merge operations*) to extract a vocabulary taking into account character level frequency and character sequences in the training corpus. This process runs up to the vocabulary limit or until the maximum defined *merge operations* is reached. An example of a training sentence after applying Truecasing and SentencePiece is shown in Fig. 3.1.

Original	`Secondly, I understand the worry of ...`
Truecase and SPM	`secondly ,_ I_ understand_ the_ wor ry_ of_ ...`

Figure 3.1: Comparison between a training sentence after applying Truecasing and SentencePiece. Note the transformation from `Secondly` to `secondly` by Truecasing and the escaping of spaces by _ and segmentation into subwords of `worry` by SentencePiece.

[12] https://www.ema.europa.eu
[12] http://bookshop.europa.eu
[12] https://multimedia.europarl.europa.eu/en
[12] https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis
[13] Whitespace is escaped by default in SentencePiece with the Unicode character ▁ (U+2581).

CHAPTER 4
Baseline Translation Systems for INTERACT-EUROPE

This chapter introduces the baseline translation systems trained for the INTERACT-EUROPE project, the followed training procedure and the experimentation done to refine their overall quality.

4.1 Baseline Models

All baseline models from this chapter were trained with Fairseq [150], a toolkit developed by the Meta AI team for the training of Seq2Seq models using the PyTorch framework [31]. In a similar vein to projects such as OpenNMT [151], Fairseq offers an extensible API with an extensive collection of tools for neural network development, with highlights such as the list of available pre-trained models and the out-of-the-box support of techniques such as multi-GPU and mixed precision training or advanced decoding algorithms. The most common way to train models in Fairseq is the usage of its CLI tooling, which also allows the use of hierarchical configuration files through Hydra [152]. The most common CLI entry points are:

- `fairseq-preprocess`: Data preprocessing and binarization for training.
- `fairseq-train`: Launching of the training with the architecture configuration and hyperparameters of the model.
- `fairseq-interactive/fairseq-generate`: Configuration and launching of the model inference.

Initial baseline models, referred to as Baseline-300M, were trained in `float16` using the "Post-LN" Transformer *Big* architecture of the original paper. A BPE SentencePiece vocabulary was trained for each language pair with a max size of 50.000 tokens and 0.9995 character coverage. For the optimizer, Adam [153] with $\beta_1 = 0.9, \beta_2 = 0.98$ was used following a inverse square root learning rate scheduler with $lr = 5e-4$ and a warm up period of 4000 steps with initial $lr = 1e-07$. Label-smoothing and dropout were respectively used with 0.1 and encoder-decoding weight embedding matrices were "tied-up" [154]. Models were trained on multiple NVIDIA's 2080 Ti GPUs through DDP for an effective batch size of 16.000 tokens, and checkpoints were kept between 10.000 updates for averaging of the last 7 to obtain the final model. Models were trained until convergence was observed on dev test or training reached one million steps. As for the decoding algorithm, beam search with size 6 was chosen for all models. Fig. A.1 in the

appendix shows the configuration of `fairseq-train` used for the training of one of these baseline models.

4.1.1. Refinements on Baseline Models

For $en \rightarrow es, de, fr$, fine-tuning (FT) was done by further training the final model with a fixed learning rate during a few iterations. For the FT data, a split was chosen made up of the complete Medline-WMT22, Europarl-ST and MuST-C [155] datasets. The data selection was made by trying to balance the close in-domain medical data with a subset of cleaner corpora that were closer to the field of spoken language.

With respect to $en \rightarrow sl$, an ablation study was made by trying an aggressive filtering with the previously mentioned Bicleaner and Bifixor,[1] the usage of CCMatrix cleaned with these tools and the addition of tagged back-translations obtained from a baseline model trained on the reverse direction with the monolingual data in Table 3.6.

In addition to this, and in order to corroborate the theory around Transformer scaling in MT [91], we experimented by training for each language pair a Pre-Norm Transformer with double the amount of layers (Baseline-600M)[2] and an increased effective batch size of 192.000 tokens, following [156]. The 1.3B variation (24x24) of the previous paper was also tried on preliminary study by training a $en \rightarrow fr$ model with FDSP, but results were found out to be slightly worse than the 600M variant. This, combined with the higher computing budget associated for training with this model size, led us to the decision of not exploring the rest of language pairs for this size. These higher variants were trained on a combination of computing nodes with NVIDIA A40s and A30s GPUs.

4.2 Evaluation of Baseline Models

Tables 4.1 and 4.2 present model evaluation with BLEU[3] and COMET-22 for each language direction. For the most part, the effect of fine-tuning these models seem relatively small, with ≤ 1 changes on both metrics. For the small $en \rightarrow fr, es$ directions, we see how the effectiveness of the fine-tuning differs if we take into account one of the chosen metrics respect the other, with BLEU indicating a worse performance of FT models respect to the baseline, while COMET indicating the opposite. As for $en \rightarrow de$, both metrics seem to indicate a boost in performance with FT.

In $en \rightarrow sl$, we observe how the initial cleaning slightly hurts performance, but the addition of CCMatrix and back-translations improves both BLEU and COMET, having the former show a boost of 0.8 points with respect to the baseline, while the latter still not closing the performance gap made by the cleaning procedure. We theorize that this discrepancy of metrics may be due to the noisier nature of CCMatrix that, alongside back-translations, may introduce unnatural phrase structures which BLEU could not penalizing as much as opposed to COMET.

Regarding the 600M parameters variants, we see a considerable improvement of both BLEU and COMET scores for $en \rightarrow fr, es, de$ when measuring against the smaller variants, both with and without fine-tuning. Interestingly, for $en \rightarrow fr$ there is a huge jump of +5 points of BLEU, but COMET-22 surprisingly seems to reflect that the bigger model is of the same quality as the smaller fine-tuned variant. For $en \rightarrow sl$, we see how the im-

[1] From 38.8M to 14.5M phrases after cleaning, $\sim$ 37% of the original data is preserved.

[2] Baseline has the same amount of layers respectively for the encoder and decoder component (6x6), so we roughly double the amount of parameters (12x12).

[3] SacreBLEU signature: `BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`

provement seems to be much smaller as opposed to the other directions, and for the case of BLEU, the CCMatrix and BT variants seem to be better in this case. Smaller BLEU variations in results for $en \rightarrow sl$ seem to indicate that the amount of training data compared to other directions may be overall hurting our ability of benefiting from model scaling.

Overviewing the results, we can affirm that the achieved translation quality are good, which are reflected in the obtained BLEU scores on the range of 40-50+, values that for this metric are normally considered in the range of high quality translations. This is also supported by the obtained COMET-22 scores, which reach the 80+ mark, a value that in the literature seems to generally correlate to similar quality translations respect those given as reference.

Table 4.1: Results of baseline $en \rightarrow fr, es, de$ models on the INTERACT dev set.

en → fr, es, de						
	French		Spanish		German	
Model	BLEU	COMET-22	BLEU	COMET-22	BLEU	COMET-22
Baseline-300M	51.5	81.8	56.4	85.4	40.9	81.5
+ FT	50.6	**82.2**	56.2	85.5	41.2	82.1
Baseline-600M	**56.4**	82.1	**58.9**	**85.9**	**43.8**	**82.4**

Table 4.2: Results of baseline $en \rightarrow sl$ models on the INTERACT dev set.

en → sl		
	Slovene	
Model	BLEU	COMET-22
Baseline-300M	40.0	84.5
+ Cleaning	39.7	83.3
+ CCMatrix	40.4	83.6
+ BT	**40.8**	84.1
Baseline-600M	40.3	**84.7**

CHAPTER 5
Adaptation of Multilingual Large Neural Models

This chapter explores the performance of publicly available multilingual pre-trained neural models respect the previous baselines of Chapter 4. Further adaptation of these models is explored to improve their performance MT, both for multilingual encoder-decoder models and decoder-only LLMs. All models from this chapter were trained through the HuggingFace Transformers [157] library.

5.1 Multilingual Encoder-Decoder Models

In the previous chapter, individual models were trained for each language pair. The problem with this methodology is that, with the increase of language directions, there is a higher deployment cost at training and inference since more models are needed to cover all directions. On the other hand, the training of a multilingual MT model (MMT) offers a great solution to this problem by unifying all directions into a unique model, thus highly reducing deployment costs. On top of this, MMT can make use of the internal representation of the model between similar languages pairs and distill knowledge, yielding better translation quality among similar language pairs. For example, MMT models have been observed to generally offer a better performance for low-resource or zero-shot language pairs [158].

However, MMT has not gained as much widespread adoption due to the additional complexity of these models and potential performance challenges that may arise. In particular, issues related to model capacity bottlenecks or poor data balancing can hinder effective generalization capabilities. When coupled with other problems such as accidental code-switching,[1] these can lead MMT to have an overall performance downgrade compared to traditional bidirectional baselines [158].

As such, during recent years the study of multilingual MT models has risen in popularity in order to minimize their problems while reaping their benefits and matching the performance of traditional bidirectional models. In this context, it is interesting to see how one of these state-of-the-art MMT models compare to our trained models and see if it can be better adapted to our domain.

[1] As in changing output language during inference.

5.1.1. No Language Left Behind

No Language Left Behind (NLLB) [159] is a series of MMT Transformer models released by the Meta research team that supports bidirectional translation of up to 200 languages. These have been observed to obtain high quality translations for the majority of language pairs which they cover. In order to maximize performance, the models were trained taking into account the quirks that may appear in MMT model training through a careful selection of techniques such as curriculum learning and data balancing. This was combined, among other things, with a complex data-mining pipeline, which stands out by its quality and scale. Respect its structure, all NLLB models follow a Pre-Norm encoder-decoder architecture along some additional modifications, such as the usage of Sparsely Activated Mixture of Experts (MoE) in its biggest variant.[2] Multilingualism is achieved by a shared SentencePiece vocabulary with special language tokens for each available language, which are added as prefixes to each part of the model: the source token with the encoder, and the target token with the decoder.

5.1.2. Experimentation on NLLB

A case study was conducted to assess the performance of the dense variants of NLLB, which range from 600M, 1.3B and 3.3B parameters, on all INTERACT dev sets. In addition to this, a collection of LoRAs were trained for each of these sizes and all language directions to study the effectiveness of this method as a lightweight domain adaptation tool. This approach aims to further specialize MMT models in a specific language direction after training. That is, by using LoRAs, possible biases in the representation space of the models resulting of multilingualism might be mitigated.

Table 5.1: LORA hyperparameters for the trained decoder models.

Hyperparameter	Value
Optimizer	AdamW [160]
Warm up Ratio	0.06
LR Schedule	Linear
Effective Batch Size	$\approx$ 16.000 tokens
Epochs	3 or until convergence
Initial Learning Rate	2e-4
Lora Dropout	0.1
Target Modules	$\boldsymbol{Q, K, V, O}$
LoRA rank config.	$r_Q = r_K = r_V = r_O = 16$
LoRA α	32
Trainable parameters[3]	0.1-0.4%

To train the LoRAs, the fine-tuning sets introduced on Sec. 4.2 were utilized for $en \rightarrow fr, es, de$, ensuring consistency with the rest of the experiments of this work. As a unique case, for $en \rightarrow sl$, we opted to utilize a random subset of medical data from the EMEA corpus and previous Oncology-Institute back-translations that approximately matched the number of sentences found in the other fine-tuning datasets. The LORA hyperparameters chosen for NLLB are presented in Table 5.1, and remained consistent across all language pairs. At evaluation time, inference hyperparameters were replicated based on those chosen for the Fairseq models of Chapter 4 alongside the use of beam search with size 6 as the decoding algorithm.

[2] In MoE, the FFN layers of the Transformer are split and activated by gates such that only a subset of model parameters, or "experts" is activated per input. For more information see §6.2 [159].

[3] Depends on the dimension and quantity of $\boldsymbol{Q, K, V, O}$ matrices, which varies between models.

Table 5.2 presents the BLEU and COMET-22 scores of the NLLB experiments along model sizes and LoRA usage. Results are divided between parameter count and language pair, as well as by usage of the corresponding trained LoRA. Baseline-600M models of Chapter 4 are also reported to serve as reference. First, in terms of scale we can see how, for all language pairs, both metrics indicate a performance boost of bigger models compared to NLLB-600M, having BLEU improvements that range from 0.5 in $en \rightarrow es$ up to 5.5 for $en \rightarrow sl$. These results are also reflected on a significant, but smaller scale, for the COMET-22 results. For non-LoRA variants, we observe how for all language pairs NLLB-600M performance is equal or worse depending on the metric, but on higher sizes and language direction results vary:

- For French, NLLB-1.3/3.3B $\leq$ Baseline-600M on BLEU and COMET.
- For Spanish, NLLB-1.3/3.3B $\sim$ Baseline-600M for BLEU, but $<$ in COMET.
- For German, NLLB-1.3/3.3B $<$ Baseline-600M for BLEU and COMET.
- For Slovene, NLLB-1.3/3.3B $\ll$ Baseline-600M for BLEU, but $>$ in COMET.

Analyzing the LoRA results reveals a significant enhancement across all models. Notably, for the majority of cases the trained LoRAs tend to exhibit substantially larger deltas in both metrics when compared to scaling model size. The most remarkable improvements in quality are observed in the 1.3B and 3.3B variants of $en \rightarrow es$, resulting in respective performance boosts of 4.0/1.5 and 2.7/1.2 points in BLEU and COMET-22. In comparison to the Baseline-600M model, the LoRA models for $en \rightarrow fr, es$ achieve performance parity for both metrics when they reach the 1.3B parameter size. Conversely, for $en \rightarrow de, sl$, BLEU scores lag behind at all sizes, but COMET-22 indicates performance gains starting from the 1.3B parameter mark.

Table 5.2: Results for NLLB models in the INTERACT dev sets with respect to the best baseline model.

		en → fr, es, de, sl							
		French		Spanish		German		Slovene	
Model	LORA	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
Baseline-600M	-	56.4	82.1	58.9	85.9	**43.8**	82.4	**40.3**	84.7
NLLB-600M	✗	53.5	82.1	55.8	85.2	38.0	81.2	33.3	83.1
NLLB-1.3B	✗	55.6	82.8	56.4	85.5	39.7	81.9	36.2	85.0
NLLB-3.3B	✗	56.4	82.8	56.3	85.6	41.5	82.0	38.8	85.2
NLLB-600M	✓	53.8	82.9	56.4	85.3	39.0	82.4	35.0	84.1
NLLB-1.3B	✓	56.5	83.8	59.1	86.7	41.1	83.4	38.8	86.3
NLLB-3.3B	✓	**57.5**	**84.3**	**60.3**	**87.1**	41.8	**83.7**	39.7	**87.0**

5.2 Multilingual Decoder-Only Models

5.2.1. BLOOM and LLAMA-2

Following the points outlined in Sec. 2.3.2 of LLMs for MT, we choose to study and experiment last years BLOOM [161] family of Transformer models, which have remained widely used by the scientific community since their release. These are a collection of open source multilingual decoder-only language models trained by the BigScience project with ROOTS [162], a massive 1.6TB corpora which covers 59 languages. In terms of architecture, these models are very similar to the base decoder of the original Transformer, with

the major differences being the usage of `bfloat16` precision, the GeLU activation function [50], ALiBI Positional Embeddings [49] and an extra normalization layer after the embeddings for better training stability. Although there exists instruction-finetuned versions of these models which offer a general performance boost, referred as BLOOMZ [157], these were not chosen due to the findings of the original paper on the degradation of quality of generation based tasks like MT.

Alongside BLOOM, and considering the apparent lack of German and Slovene data in the ROOTS corpus, the evaluation of the recently released LLAMA-2 models [163] published by Meta, was also chosen as an object of our study. These are an improvement of the LLAMA models released earlier this year [164] by the same team. They were trained with over 40% more tokens and double the context window compared to the original versions, as well as a variation of the attention mechanism called Grouped-Query Attention (GQA) [165]. In terms of architecture, both the LLAMA and LLAMA-2 follow closely the decoder-only approach, with only three major differences: the usage of Pre-Norm with RMSNorm [166], the SwiGLU activation function [167] and Rotary Positional Embeddings [48]. In the same way as BLOOM, there exists variants of LLAMA-2 fine-tuned with instruction datasets, in this case with Reinforcement Learning With Human Feedback (RLHF) [168], which is currently a popular technique for aligning LLMs to desired preferences and behaviors. These, however, were left out from this study due to the nature of being biased to chat assistant applications, which for the case of MT on the INTERACT domain, may generate unnecessary remarks regarding its role as an assistant and end up hurting performance.

5.2.2. Experiments on decoder models

The In-Context Learning capabilities of both BLOOM and LLAMA-2 are evaluated by varying the amount of translation pair shots on model input from 0 to 3. Alongside this, we consider the usage of LoRAs to see which methodology or combination obtains better downstream task performance. The previous fine-tuning datasets from Chapter 4 and EMEA/Oncology-BT mix from Sec. 5.1.2 are used for both shot selection and LORA fine-tuning datasets. In case of the former, a preliminary study was made for smart prompt selection through the retrieval of the k-best-shots of a FAISS [169] index built from the training data. In it, cosine similarity between embeddings obtained from the sentence embedding model LaBSE [170] on source phrases was used. Results on this strategy were ultimately found to have no statistical significance on evaluation metrics when compared to randomly selecting input prompts. LoRAs were trained with the same hyperparameters of Table 5.1. Inference hyperparameters where kept consistent to those selected for the NLLB models, with the exception of the restriction of forcing the model to not repeat any previously generated 6-grams, which helped alleviate word repetition problems that were observed in preliminary experiments.

```
[PROMPT]       →  [INSTRUCTION]([SHOT])*

[INSTRUCTION] →  Translate from {src_name} to {tgt_name}:\n
([SHOT])*      →  {src_name}: {src_phrase} = {tgt_name}: {tgt_phrase}\n
```

Figure 5.1: Prompt format for decoder models. Square brackets tags are not present on input, and are provided for better context on overall prompt structure.

All data was formatted to follow the prompt structure of Fig. 5.1, which can be divided between an `[INSTRUCTION]` header and one or more `[SHOT]` translation phrases. Each curly braces pair, such as `src_name`, indicates a substitution by their corresponding values. For the source and target names, these were given by transforming input language codes to the corresponding names from the `langcodes`[4] library. As indicated by the regex style mark on Fig. 5.1, possible k-shots translation examples were introduced with the shot format phrase after the instruction header and before the last shot, which acted as the input phrase. During training and inference, the `tgt_phrase` variable acted as the target labels, leaving the rest of the prompt as input. Due to memory constraints, no additional shots were provided to the model during training except the phrase to be translated.

On the basis of previous results with BLOOM on MT [81], and in order for comparisons to be fair with respect the previous studied models and realistic resource contained scenarios, we mainly restrict the study to the experimentation with the 3.3B variation of BLOOM and the smallest variant of LLAMA-2 with 7B, which although has higher parameter count, we found had similar speed and memory usage at inference time, which we attribute to the usage of the GQA mechanism. We additionally try the biggest model size of BLOOM, which reaches the 175B parameter count, for $en \rightarrow es, fr$ to test the limits of LLMs in MT without limitations. For the inference of this bigger model, we map weights along 8 A40 GPUs, quantize model parameters to `int8`, CPU-offload part of them and make use of greedy-decoding. While quantization often introduces a performance hit, BLOOM was found to be relatively more robust than other LLMs when it comes to post-training quantization [171].

Table 5.3: Results for NLLB in the INTERACT dev sets with respect to the best baseline model.

			en → fr, es, de, sl							
			French		Spanish		German		Slovene	
Model	Shots	LORA	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
Baseline-600M	-	-	**56.4**	82.1	**58.4**	**85.9**	**43.8**	82.4	**40.3**	**84.7**
BLOOM-3B	0	✗	12.6	65.6	35.9	78.3	5.8	54.0	0.7	39.6
	1		36.8	78.9	41.1	80.0	9.3	53.0	0.5	40.8
	2		38.0	77.0	40.2	79.2	8.3	50.8	0.2	40.8
	3		39.4	79.0	39.8	78.2	7.1	48.9	0.0	41.0
	0	✓	45.5	82.1	52.0	85.3	25.6	73.6	7.6	55.4
LLAMA-2-7B	0	✗	18.2	64.8	15.9	64.7	13.2	66.7	8.9	55.7
	1		32.6	75.3	34.2	74.5	18.9	72.4	12.9	67.0
	2		34.8	75.7	41.0	81.7	22.7	74.4	15.0	70.7
	3		35.3	75.6	42.4	82.7	24.6	75.6	15.6	72.7
	0	✓	48.5	**82.7**	52.0	85.6	34.6	**82.4**	21.9	81.9
BLOOM-175	0	✗	45.3	80.1	45.2	82.7	-		-	
	1		48.5	82.2	50.3	82.9				

Table 5.3 shows results for each language pair across k-shots and LoRA usage. In the case of $en \rightarrow fr, es, de$ we see very similar results with both BLOOM-3B and LLAMA-2-7B showing a considerable jump of quality for both metrics passing from 0 to 1-shot, with smaller, but significant smaller jumps for 2 and 3-shots. The only exceptions to this rule are the respective $en \rightarrow es, de$ BLOOM-3B models, which plateau at 1-shot, having slight performances hits when increasing the shot amount. For the $en \rightarrow fr$ direction, when shots are provided, BLOOM-3B works better than LLAMA-2-7B, while for $en \rightarrow es, de$ the opposite is true. Regarding the results of BLOOM-175B, 0-shot and 1-shot performance is considerably higher at 175B when compared to its smaller variant, specially for the $en \rightarrow fr$ direction.

[4] https://pypi.org/project/langcodes/

As for $en \rightarrow sl$, we see how BLOOM-3B fails completely at the MT task. LLAMA-2-7B, on the other hand, is able to obtain translations, although they are still relatively worse than the results of the baseline models and NLLB.[5] For base BLOOM-3B, we can also observe relatively low quality scores for $en \rightarrow de$. This behavior of BLOOM can be attributed to the previously mentioned lack of German and Slovene data of the ROOTS corpus.

Regarding the trained LoRAs, we see a major boost in performance with increases as high of +10 points of BLEU for the majority of models and directions, with similar COMET-22 jumps in the range of 3-7 points. For $en \rightarrow fr, de, sl$ LLAMA-2-7B LoRAs outperform their BLOOM-3B counterparts, while in $en \rightarrow es$ both methods have the same BLEU scores, with LLAMA-2-7B being slightly better by 0.3 points of COMET-22.

Compared to the Baseline-600M models, we see how our trained LLMs seem to be not as competitive in terms of BLEU, but the LLAMA-2-7B LoRAs close the gap considerably when looking at COMET-22 results for $en \rightarrow fr, de$. We can see also observe how both BLOOM-3B and LLAMA-2-7B LoRAs are able to match performance of the BLOOM-175B model.

Interestingly, when increasing k-shots with LoRA we observe that performance is degraded and fluctuates greatly. This behavior can be seen in Tab.A.1 of Appendix A, and we attribute it to a biasing of the model to the input prompt format being 0-shot during training, overwriting possible ICL capabilities. We also observe for the previous language pairs that hallucinations typical in the usage of MT LLMs [86] decrease considerably for our models when jumping from 0-shot to 1-shot, and mostly disappear when using LoRA with no shots.

5.3 Evaluation on INTERACT-EUROPE test sets

In order to compare results of all trained models of this work, we report results on the INTERACT-EUROPE test sets. Table 5.4 summarizes the performance of the best models of each model category: The encoder-decoders of the bilingual baselines (Baseline-600M) and the multilingual LLMs (NLLB-600M/3.3B), as well as the best decoder-only configurations (LLAMA-2-7B). For NLLB and LLAMA, LoRA models are selected for $en \rightarrow fr, es, de, sl$. In the case of NLLB, we choose to report both 600M and 3.3B to have comparisons that are more fair to the parameter count of Baseline-600M and LLAMA-2-7B.

Table 5.4: Results for best trained models of each architecture in the INTERACT test set.

	en → fr, es, de, sl							
	French		Spanish		German		Slovene	
Name	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
Baseline-600M	**56.0**	**84.2**	58.8	86.9	**43.3**	83.4	**39.9**	85.9
NLLB-600M	50.1	81.3	56.8	86.7	39.4	83.2	34.1	85.1
NLLB-3B	52.5	82.0	**59.8**	**87.7**	42.7	**84.9**	39.4	**86.6**
LLAMA-2-7B	47.5	83.9	51.9	86.3	34.7	83.7	20.4	80.2

In all language pairs, performance of Baseline-600M is better than NLLB-600M in both BLEU and COMET-22 scores. As for NLLB-3B, in $en \rightarrow fr$ the baseline models scores are still superior by a considerable margin of 3.5/2.2 points, in $en \rightarrow es$ NLLB

[5] Respect our translation directions, the total language distribution of the pre-training data of LLAMA-2 is: English | 89.70%, German | 0.17%, French | 0.16% Spanish | 0.13%, Slovene | 0.01%.

scores are higher in both metrics by 1.0/0.8 points and in $en \rightarrow de, sl$ the Baseline-600M slightly outperforms in terms of BLUE, but worsens in COMET-22. As for the LLAMA-2-7B, performance of $en \rightarrow sl$ is poor, and BLEU of the rest of languages is lower than that of the rest of models. However, if we look at the COMET-22 scores, they reach similar or even better values for some cases of $en \rightarrow fr, es, de$. We theorize that this contradictory behavior between BLEU and COMET-22 scores are due to the previously mentioned nature of LLMs translations on Sec. 2.3.2 and its interaction with MT metrics.

5.3.1. Evaluation considering computing constraints

In the context of the INTERACT-EUROPE project, where these models are being deployed on for both offline and online usage in real-world applications, it is crucial to not only take into account the previously reported differences between evaluation metrics, but consider latency, possible computing limitations and overall generation stability of these models. For each type of model, we can distinguish a series of advantages and disadvantages. For the case of the bilingual baselines, we overall find the best performance and speed inference trade-off for all languages pairs, but at the same time there is considerable overhead in data collection, model training and lastly, memory usage if multiple language directions need to be translating at the same time. As for the pre-trained multilingual models, we find that for the proper MT based encoder-decoder models we could easily extend them to other European language directions that may be interesting to the project, but performance parity with Baseline models could only be reached by taking the small training overhead of PEFT and a 5.5x increase in size, the latter being reflected in an approximate two times slower inference time. Regarding the decoder-only models, we have overall more flexibility, such as in the ways we could control the resulting MT through ICL or extend the resulting model to other related language tasks, but looking at the results we cannot be sure of the stability of the models without further testing, especially if LoRA is not used and were only left ICL, which gets very expensive as the number of shots increases.

Taking these points into account, we highlight the following observations: First, that if you have the computing power, data and time, bilingual models are still a solid choice for training a reliable NMT model, specially if offering a high amount of translations directions is not of your interest. Second, that by assuming a slight trade off in speed, pre-trained multilingual models can further achieve similar performance to bilingual baselines. Third, that PEFT is a fast and efficient way to further improve general model performance. And last, that in a similar way to the results that have been observed in the literature, decoder LLMs are adaptable and can achieve solid results in MT, but the stability, necessity of model size for effectiveness on the task and uncertainty of translation quality, leaves these models with still much to be desired for them to be established as the go-to choice when building a MT system.

CHAPTER 6
Conclusions

This chapter serves as a comprehensive summary of all the tasks undertaken in this study, aligning them with the objectives initially outlined in Sec. 1.3. Additionally, we delve into the conclusions drawn from these endeavors and explore possible avenues for future research.

Chapter 1 has laid the foundation for this work by introducing the overarching framework and essential theoretical fundamentals within the field of Machine Learning. In Chapter 2, we have taken a thorough examination of the theories and tools pertinent to NMT, with a particular emphasis on the Transformer architecture and the developments surrounding the so called LLMs and the ways in which these are adapted into NMT.

Chapter 3 has introduced the data used for the model training and evaluation used throughout the work, and Chapter 4 has described the subsequent training and evaluation of a series of bilingual NMT models on the language pairs of the INTERACT-EUROPE project. Lastly, in Chapter 5 we have delved in the adaptation of different multilingual pretrained models through different methodologies to the MT task. Here, as closing remarks we have presented a comprehensive evaluation of the overall performance of each best model per trained architecture on the INTERACT-EUROPE test set, with an intricate analysis on the possible strengths and weaknesses when applying the models on real-world scenarios.

Regarding the proposed objectives, we can affirm that each one has been properly addressed on throughout our work. Our best trained model across all language pairs featured in the INTERACT-EUROPE datasets are performance-wise on pair with other available state-of-the-art large models on MT. In addition to this we have explored publicly available models in the form of NLLB, BLOOM and LLAMA. Lastly, we made an study on the adaptation of LLMs for the MT task by considering the latest advancements and techniques of the field.

Concerning future work, several areas warrant additional exploration and consideration. With the clear performance boost with model scale, further research in optimizing time and size constraints of general purpose Transformers and LLMs usage with PEFT methods and model quantization is key. In this context, the exploration of the recently line of work on the construction of a non-parametric datastore via k-Nearest-Neighbor retrieval [172], [173], [174], [175] makes for an interesting way to cheaply adapt models to further domains and language pairs. Further work for adaptability of LLMs on real-world scenarios, such as part of end-to-end Speech Translation pipeline or a Interactive MT system, also make for a interesting proposal to undertake with the lessons learned from this work. Lastly, additional exploration on performance trade-offs between bilingual and multilingual MT models needs to be considered in future work after the observed results.

Acknowledgments

In the context of the **INTERACT-EUROPE** project, the research leading to these results has received funding from EU4Health Programme 2021-2027 as part of Europe's Beating Cancer Plan under Grant Agreement nº101056995. Additionally, the author would like to thank the **ValgrAI-Valencian Graduate School and Research Network for Artificial Intelligence** and **Generalitat Valenciana** for their economical support in the form of a grant for his Master's Studies, which this dissertation is a part of. In a similar vein, the author would like to thank the **Valencian Research Institute for Artificial Intelligence** for their generosity in allowing access to their internal computing cluster and the rest of my colleagues from the **Machine Learning and Language Processing** group.

Bibliography

[1] Innovative collaboration for Inter-specialty cancer training across Europe Interact-Europe project. https://www.europeancancer.org/eu-projects/impact/interact-europe.

[2] Jorge Iranzo Sánchez. *Sistemas de síntesis de voz basados en redes neuronales para lenguas europeas*. PhD thesis, Universitat Politècnica de València, 2022.

[3] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[4] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

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

[6] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814. Omnipress, 2010.

[7] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *CoRR*, abs/1912.05911, 2019.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734. ACL, 2014.

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

[11] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.

[12] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML (3)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1310–1318. JMLR.org, 2013.

[13] Peter Toma. Systran as a multilingual machine translation system. In *Proceedings of the Third European Congress on Information Systems and Networks, overcoming the language barrier*, pages 569–581, 1977.

[14] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

[15] Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguistics*, 16(2):79–85, 1990.

[16] Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguistics*, 19(2):263–311, 1993.

[17] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003.

[18] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[19] Asunción Castaño, Francisco Casacuberta, and Enrique Vidal. Machine translation using neural networks and finite-state models. In *Proceedings of the 7th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, St John's College, Santa Fe, July 23-25 1997.

[20] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6:40–53, 03 2008.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[22] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[24] Amirhossein Kazemnejad. Transformer architecture: The positional encoding. *kazemnejad.com*, 2019.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[26] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[27] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018.

[28] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR, 2020.

[29] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 3:111–132, 2022.

[30] Martin Popel and Ondrej Bojar. Training tips for the transformer model. *Prague Bull. Math. Linguistics*, 110:43–70, 2018.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[32] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12):3005–3018, 2020.

[33] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, and Shen Li. Pytorch FSDP: experiences on scaling fully sharded data parallel. *CoRR*, abs/2304.11277, 2023.

[34] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In *SC*, page 20. IEEE/ACM, 2020.

[35] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *ICLR (Poster)*. OpenReview.net, 2018.

[36] Dhiraj D. Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. A study of BFLOAT16 for deep learning training. *CoRR*, abs/1905.12322, 2019.

[37] Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart F. Oberman, Mohammad Shoeybi, Michael Y. Siu, and Hao Wu. FP8 formats for deep learning. *CoRR*, abs/2209.05433, 2022.

[38] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. *CoRR*, abs/2208.07339.

[39] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs.

[40] Yichi Zhang, Ankush Garg, Yuan Cao, Łukasz Lew, Behrooz Ghorbani, Zhiru Zhang, and Orhan Firat. Binarized neural machine translation.

[41] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *WMT*, pages 1–9. Association for Computational Linguistics, 2018.

[42] Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas, November 2016. Association for Computational Linguistics.

[43] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics.

[44] Franck Burlot and François Yvon. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[45] Benjamin Marie, Raphael Rubino, and Atsushi Fujita. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, Online, July 2020. Association for Computational Linguistics.

[46] Martin Gellerstam. Translationese in swedish novels translated from english. 1986.

[47] Nikolay Bogoychev and Rico Sennrich. Domain, translationese and noise in synthetic data for neural machine translation. *CoRR*, abs/1911.03362, 2019.

[48] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864, 2021.

[49] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*. OpenReview.net, 2022.

[50] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[51] Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodola. Accelerating transformer inference for translation via parallel decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12336–12355, Toronto, Canada, July 2023. Association for Computational Linguistics.

[52] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR, 23–29 Jul 2023.

[53] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022.

[54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[55] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040, 2020.

[56] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *AAAI*, pages 6706–6713. AAAI Press, 2019.

[57] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers. *CoRR*, abs/2301.02111, 2023.

[58] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[59] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022.

[60] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*. OpenReview.net, 2022.

[61] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran G. V., Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: reinventing rnns for the transformer era. *CoRR*, abs/2305.13048, 2023.

[62] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang,

Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *CoRR*, abs/2303.18223.

[63] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[64] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[65] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[66] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *ArXiv*, abs/2301.00234, 2022.

[67] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[68] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[69] Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. Are emergent abilities in large language models just in-context learning?, 2023.

[70] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage?, 2023.

[71] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *CoRR*, abs/2212.07677, 2022.

[72] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada, July 2023. Association for Computational Linguistics.

[73] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *ICLR*. OpenReview.net, 2023.

[74] Shuo Wang, Zhaopeng Tu, Zhixing Tan, Wenxuan Wang, Maosong Sun, and Yang Liu. Language models are good translators. *CoRR*, abs/2106.13627, 2021.

[75] Yingbo Gao, Christian Herold, Zijian Yang, and Hermann Ney. Is encoder-decoder redundant for neural machine translation? In Yulan He, Heng Ji, Yang Liu, Sujian Li, Chia-Hui Chang, Soujanya Poria, Chenghua Lin, Wray L. Buntine, Maria

Liakata, Hanqi Yan, Zonghan Yan, Sebastian Ruder, Xiaojun Wan, Miguel Arana-Catania, Zhongyu Wei, Hen-Hsen Huang, Jheng-Long Wu, Min-Yuh Day, Pengfei Liu, and Ruifeng Xu, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 562–574. Association for Computational Linguistics, 2022.

[76] Biao Zhang, Behrooz Ghorbani, Ankur Bapna, Yong Cheng, Xavier Garcia, Jonathan Shen, and Orhan Firat. Examining scaling and transfer of language model architectures for machine translation. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 26176–26192. PMLR, 2022.

[77] Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. Multilingual machine translation with large language models: Empirical results and analysis. *ArXiv*, abs/2304.04675, 2023.

[78] Amr Hendy, Mohamed Gomaa Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. How good are gpt models at machine translation? a comprehensive evaluation. *ArXiv*, abs/2302.09210, 2023.

[79] Yuan Gao, Ruili Wang, and Feng Hou. Unleashing the power of chatgpt for translation: An empirical study. *ArXiv*, abs/2304.02182, 2023.

[80] Yasmin Moslem, Rejwanul Haque, and Andy Way. Adaptive machine translation with large language models. *ArXiv*, abs/2301.13294, 2023.

[81] Rachel Bawden and Franccois Yvon. Investigating the translation performance of a large multilingual language model: the case of bloom. *ArXiv*, abs/2303.01911, 2023.

[82] Jiahuan Li, Hao Zhou, Shujian Huang, Shan Chen, and Jiajun Chen. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions. *ArXiv*, abs/2305.15083, 2023.

[83] Biao Zhang, Barry Haddow, and Alexandra Birch. Prompting large language model for machine translation: A case study. *ArXiv*, abs/2301.07069, 2023.

[84] Xavier García, Yamini Bansal, Colin Cherry, George F. Foster, Maxim Krikun, Fan Feng, Melvin Johnson, and Orhan Firat. The unreasonable effectiveness of few-shot learning for machine translation. *ArXiv*, abs/2302.01398, 2023.

[85] Vikas Raunak, Arul Menezes, Matt Post, and Hany Hassan. Do gpts produce less literal translations? In *ACL (2)*, pages 1041–1050. Association for Computational Linguistics, 2023.

[86] Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André Martins. Hallucinations in large multilingual translation models. *ArXiv*, abs/2303.16104, 2023.

[87] Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online, November 2021. Association for Computational Linguistics.

[88] Yifan Wang, Zewei Sun, Shanbo Cheng, Weiguo Zheng, and Mingxuan Wang. Controlling styles in neural machine translation with activation prompt. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2606–2620, Toronto, Canada, July 2023. Association for Computational Linguistics.

[89] Eleftheria Briakou, Colin Cherry, and George F. Foster. Searching for needles in a haystack: On the role of incidental bilingualism in palm's translation capability. *CoRR*, abs/2305.10266, 2023.

[90] Mitchell A. Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5915–5922. Association for Computational Linguistics, 2021.

[91] Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation, 2021.

[92] Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[93] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019.

[94] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022.

[95] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *ICLR (Poster)*. OpenReview.net, 2018.

[96] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online, August 2021. Association for Computational Linguistics.

[97] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL, 2002.

[98] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[99] Maja Popović. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[100] Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *AMTA*, pages 223–231. Association for Machine Translation in the Americas, 2006.

[101] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[102] Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *arXiv preprint arXiv:2107.10821*, 2021.

[103] Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George F. Foster, Alon Lavie, and André F. T. Martins. Results of WMT22 metrics shared task: Stop using BLEU - neural metrics are better and more robust. In *WMT*, pages 46–68. Association for Computational Linguistics, 2022.

[104] Ricardo Rei, José G. C. de Souza, Duarte M. Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luísa Coheur, and André F. T. Martins. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *WMT*, pages 578–585. Association for Computational Linguistics, 2022.

[105] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *ICLR*. OpenReview.net, 2020.

[106] Nuno M. Guerreiro, Elena Voita, and André F. T. Martins. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2022.

[107] Duarte Alves, Ricardo Rei, Ana C Farinha, José G. C. de Souza, and André F. T. Martins. Robust MT evaluation with sentence-level multilingual augmentation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 469–478, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[108] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*, 2020.

[109] Amy Pu, Hyung Won Chung, Ankur P Parikh, Sebastian Gehrmann, and Thibault Sellam. Learning compact metrics for mt. In *Proceedings of EMNLP*, 2021.

[110] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics.

[111] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[112] Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy, August 2019. Association for Computational Linguistics.

[113] Terry Yue Zhuo, Qiongkai Xu, Xuanli He, and Trevor Cohn. Rethinking round-trip translation for machine translation evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 319–337, Toronto, Canada, July 2023. Association for Computational Linguistics.

[114] Patrick Wilken, Panayota Georgakopoulou, and Evgeny Matusov. Suber - A metric for automatic evaluation of subtitle quality. In *IWSLT@ACL*, pages 1–10. Association for Computational Linguistics, 2022.

[115] Billy T. M. Wong and Chunyu Kit. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1060–1068, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[116] Yuchen Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Jian Yang, Haoyang Huang, Rico Sennrich, Ryan Cotterell, Mrinmaya Sachan, and Ming Zhou. BlonDe: An automatic evaluation metric for document-level machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1550–1565, Seattle, United States, July 2022. Association for Computational Linguistics.

[117] Giorgos Vernikos, Brian Thompson, Prashant Mathur, and Marcello Federico. Embarrassingly easy document-level MT metrics: How to convert any pretrained metric into a document-level metric. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 118–128, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[118] Brian Thompson and Philipp Koehn. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China, November 2019. Association for Computational Linguistics.

[119] Jörg Tiedemann and Lars Nygaard. The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC*. Citeseer, 2004.

[120] Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, 2020.

[121] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86, 2005.

[122] Mariana Neves, Antonio Jimeno Yepes, Amy Siu, Roland Roller, Philippe Thomas, Maika Vicente Navarro, Lana Yeganova, Dina Wiemann, Giorgio Maria Di Nunzio, Federica Vezzani, Christel Gerardin, Rachel Bawden, Darryl Johan Estrada, Salvador Lima-lopez, Eulalia Farre-maduel, Martin Krallinger, Cristian Grozea, and

Aurelie Neveol. Findings of the WMT 2022 biomedical translation shared task: Monolingual clinical case reports. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 694–723, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[123] Andrej Pancur and Tomaž Erjavec. The siParl corpus of Slovene parliamentary proceedings. In *Proceedings of the Second ParlaCLARIN Workshop*, pages 28–34, Marseille, France, May 2020. European Language Resources Association.

[124] Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online, August 2021. Association for Computational Linguistics.

[125] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[126] Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium, October. Association for Computational Linguistics.

[127] Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal, November 2020. European Association for Machine Translation.

[128] Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, 2016.

[129] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer, 2012.

[130] Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1850–1855, 2014.

[131] Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Association for Computational Linguistics, November 2020.

[132] Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. Dgt-tm: A freely available translation memory in 22 languages. *arXiv preprint arXiv:1309.5226*, 2013.

[133] Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*, 2019.

[134] Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerda, Javier Jorge, Nahuel Roselló, Adria Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233. IEEE, 2020.

[135] Andreas Eisele and Yu Chen. MultiUN: A multilingual corpus from united nation documents. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[136] Krzysztof Wołk and Krzysztof Marasek. Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs. *Procedia Technology*, 18:126–132, 2014. International workshop on Innovations in Information and Communication Science and Technology, IICST 2014, 3-5 September 2014, Warsaw, Poland.

[137] Roberts Rozis and Raivis Skadins. Tilde MODEL - multilingual open data for EU languages. In *NODALIDA*, volume 131 of *Linköping Electronic Conference Proceedings*, pages 263–265. Linköping University Electronic Press / Association for Computational Linguistics, 2017.

[138] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia, editors. *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[139] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics.

[140] Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online, November 2020. Association for Computational Linguistics.

[141] Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[142] Roberts Rozis and Raivis Skadins. Tilde MODEL - multilingual open data for EU languages. In *NODALIDA*, volume 131 of *Linköping Electronic Conference Proceedings*, pages 263–265. Linköping University Electronic Press / Association for Computational Linguistics, 2017.

[143] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020.

[144] Mauro Cettolo, Christian Girardi, and Marcello Federico. WIT3: web inventory of transcribed and translated talks. In *EAMT*, pages 261–268. European Association for Machine Translation, 2012.

[145] Lucian Vlad Lita, Abraham Ittycheriah, Salim Roukos, and Nanda Kambhatla. truecasing. In *ACL*, pages 152–159. ACL, 2003.

[146] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[147] Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, feb 1994.

[148] 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, August 2016. Association for Computational Linguistics.

[149] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL (1)*, pages 66–75. Association for Computational Linguistics, 2018.

[150] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics, 2019.

[151] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[152] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019.

[153] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[154] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *EACL (2)*, pages 157–163. Association for Computational Linguistics, 2017.

[155] Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[156] Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. Facebook ai's WMT21 news translation task submission. In *WMT@EMNLP*, pages 205–215. Association for Computational Linguistics, 2021.

[157] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In *ACL (1)*, pages 15991–16111. Association for Computational Linguistics, 2023.

[158] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5), sep 2020.

[159] Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

[160] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019.

[161] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022.

[162] Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Sasko, Quentin Lhoest, Angelina McMillan-Major, Gérard Dupont, Stella Biderman, Anna Rogers, Loubna Ben Allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Ifeoluwa Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Luccioni, and Yacine Jernite. The BigScience ROOTS corpus: A 1.6tb composite multilingual dataset. *CoRR*, abs/2303.03915.

[163] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao,

Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[164] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

[165] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: training generalized multi-query transformer models from multi-head checkpoints. *CoRR*, abs/2305.13245, 2023.

[166] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, pages 12360–12371, 2019.

[167] Noam Shazeer. GLU variants improve transformer. *CoRR*, abs/2002.05202, 2020.

[168] Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*, 2022. https://huggingface.co/blog/rlhf.

[169] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[170] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In *ACL (1)*, pages 878–891. Association for Computational Linguistics, 2022.

[171] Arash Ahmadian, Saurabh Dash, Hongyu Chen, Bharat Venkitesh, Stephen Gou, Phil Blunsom, A. Ustun, and Sara Hooker. Intriguing properties of quantization at scale.

[172] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *ICLR*. OpenReview.net, 2020.

[173] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In *ICLR*. OpenReview.net, 2021.

[174] Yichao Du, Weizhi Wang, Zhirui Zhang, Boxing Chen, Tong Xu, Jun Xie, and Enhong Chen. Non-parametric domain adaptation for end-to-end speech translation. In *EMNLP*, pages 306–320. Association for Computational Linguistics, 2022.

[175] Hao Zhang, Nianwen Si, Yaqi Chen, Wenlin Zhang, Xukui Yang, Dan Qu, and Zhen Li. Decouple non-parametric knowledge distillation for end-to-end speech translation, 2023.

APPENDIX A
Additional experiments information

Fairseq CLI parameters

Fig.A.1 shows the list of Fairseq CLI options used when training baseline models.

```
--arch transformer_vaswani_wmt_en_fr_big \
--share-all-embeddings \
--optimizer adam \
--adam-betas '(0.9, 0.98)' \
--clip-norm 0.0 \
--lr-scheduler inverse_sqrt \
--warmup-init-lr 1e-07 \
--warmup-updates 4000 \
--lr 0.0005 \
--min-lr 1e-09 \
--dropout 0.1 \
--weight-decay 0.0 \
--criterion label_smoothed_cross_entropy \
--label-smoothing 0.1 \
--max-tokens 2000 \
--update-freq 8 \
--log-interval 100 \
--max-update 1000000 \
--max-source-positions 250 \
--max-target-positions 250 \
--fp16
```

Figure A.1: Flags and hyperparameters indicated to `fairseq-train` relevant to the architecture and training of a baseline model (Transformer *Big*).

ICL degradation results with LoRA

Table A.1 shows the BLEU and COMET-22 scores of LoRA across 1-3 shots and all language pairs, where we can observe the ICL degradation previously discussed on Sec.5.2.2.

Table A.1: Results for decoder models in the INTERACT dev set for the 1-3 shots LoRA variants of the trained decoder LLMs.

			French		Spanish		German		Slovene	
Modelo	shots	LORA	BLEU	COMET-22	BLEU	COMET-22	BLEU	COMET-22	BLEU	COMET-22
BLOOM-3B	1		38.1	78.0	38.2	73.1	8.0	48.1	4.8	49.7
	2	✗	36.1	80.3	38.1	74.2	8.2	49.1	4.6	47.0
	3		32.9	78.6	41.9	77.7	8.0	50.4	3.0	45.6
LLAMA-2-7B	1		32.1	70.3	50.2	84.3	32.3	80.2	21.3	80.2
	2	✗	20.3	59.5	47.8	47.8	30.5	77.2	20.0	73.9
	3		26.9	62.8	45.6	45.6	29.3	76.8	19.3	72.3

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				**X**
ODS 2. **Hambre cero.**				**X**
ODS 3. **Salud y bienestar.**		**X**		
ODS 4. **Educación de calidad.**	**X**			
ODS 5. **Igualdad de género.**				**X**
ODS 6. **Agua limpia y saneamiento.**				**X**
ODS 7. **Energía asequible y no contaminante.**				**X**
ODS 8. **Trabajo decente y crecimiento económico.**			**X**	
ODS 9. **Industria, innovación e infraestructuras.**		**X**		
ODS 10. **Reducción de las desigualdades.**		**X**		
ODS 11. **Ciudades y comunidades sostenibles.**				**X**
ODS 12. **Producción y consumo responsables.**				**X**
ODS 13. **Acción por el clima.**				**X**
ODS 14. **Vida submarina.**				**X**
ODS 15. **Vida de ecosistemas terrestres.**				**X**
ODS 16. **Paz, justicia e instituciones sólidas.**				**X**
ODS 17. **Alianzas para lograr objetivos.**			**X**	

Reflexión sobre la relación del TFG con los ODS más relacionados.

Es muy importante analizar la contribución que se hace cuando se realizan trabajos científicos y las mejoras que se pretenden alcanzar con estos.

A partir de los Objetivos de Desarrollo Sostenibles elaborados por la ONU en el marco de la Agenda 2030 (aprobados en 2015), se han relacionado los siguientes ODS como los más relevantes respecto a los objetivos del proyecto:

- **Educación de calidad (ODS 4):** La traducción automática es una herramienta poderosa para la mejora de la calidad educativa. Este ODS está estrechamente relacionado con este TFM, dado que el trabajo de investigación desarrollado dentro del proyecto **INTERACT-EUROPE** tiene el obetivo final de facilitar el acceso de profesionales medicos a contenido educativo que no esté en su lengua nativa.

- **Industria, innovación e infraestructuras (ODS 9):** En el trabajo se trata el aprendizaje automático, que con los recientes avances se ha establecido como una de las áreas líderes en innovación tecnológica, recibiendo multitud de usos en la industria y en el día a día. Desde los últimos años, el interés de esta área no hace más que aumentar a medida que siguen apareciendo más y más aplicaciones que integran esta tecnología en su flujo de trabajo diario. La investigación de las capacidades de LLMs es extremadamente relevante en este caso.

- **Reducción de las desigualdades (ODS 10):** El uso de modelos de traducción automática en este trabajo tiene un especial interés para crear sistemas que logren romper las barreras lingüísticas y de comunicación. La traducción automática, junto a otras tecnologías del procesamiento natural del lenguaje como el ASR o el TTS, pueden usarse en conjunto para crear productos con gran relevancia en el día a día.

- **Salud y bienestar (ODS 3):** Los modelos han sido entrenados en el contexto del proyecto **INTERACT** para su uso dentro del campo oncológico para facilitar el acceso a información a individuos, y que por ende, puedan proporcionar mejores cuidados a pacientes.

Además de estos, se ha identificado que los ODS de **Trabajo decente y crecimiento económico (ODS 8)** y **Alianzas para lograr objetivos (ODS 17)** también pueden aplicarse a este trabajo. En el caso del **ODS 8**, el uso de la traducción automatica por profesionales médicos también puede facilitar el crecimiento económico agilizando la comunicación y cooperación en el ámbito laboral de equipos con miembros en distintos países. Por último, en relación con el **ODS 17**, la traducción automática puede facilitar la comunicación dentro de las instituciones medícas y gubernamentales de distintos países.

Respecto al resto de ODS, se ha considerado que no proceden, dado que o bien no están relacionados con el área de investigación de este trabajo, o bien el uso de la tecnología propuesta no ha podido extenderse para cubrirlos.