Alma Mater Studiorum · Università di Bologna

DIPARTIMENTO DI INTERPRETAZIONE E TRADUZIONE
Corso di Laurea magistrale in Specialized Translation (classe LM-94)

TESI DI LAUREA
in COMPUTATIONAL LINGUISTICS

Return to the Source: Assessing Machine Translation Suitability based on the Source Text using XLM-RoBERTa

CANDIDATO:
Francesco Fernicola

RELATORE:
Alberto Barrón-Cedeño

CORRELATRICE:
Silvia Bernardini

CORRELATORE:
Federico Garcea

Anno Accademico 2020/2021
Terzo Appello

Acknowledgements

I want to express my utmost gratitude to all the people who supported me and made me reach this milestone.

Thanks to Prof. Alberto Barrón-Cedeño, prof. Silvia Bernardini and Federico Garcea for their dedication to this work and all the feedback they provided during these last few months. Thanks to prof. Adriano Ferraresi for his valuable insights on this dissertation and for first introducing me to the world of research.

Thanks to Francisco Guzmán for his precious counsel and for directing my attention towards the latest developments in Quality Estimation metrics.

Thanks to Alex Yanishevsky for our discussions on the impact of the source text on machine translation after our meeting at the MTSummit2021.

Thanks to Antonio, Marghe and Stefan for these two years. Forlí knows.

Thanks to the Sala San Luigi for the countless movies that lightened my Tuesdays.

Thanks to Giovanni, to whom I now officially owe an amaro.

Thanks to the Beegs and their incomparable humor.

Thanks to all the people that crossed my path and supported me from Caserta, to Forlì, Saarbrücken and Hamburg. Wherever you are now in the world, I hope you are safe and I am sure someday we will be able to once again laugh together.

Last but not least, thanks to my family.

So long, [Forlì] *and thanks for all the fish.*

Abstract

In order to assess the suitability of a text for machine translation (MT), the factors in play are many and often vary across language pairs. Readability might certainly account for part of the problem, but the metrics for its evaluation are inherently monolingual (e.g., Gunning fog index) or have language learning as a target. Thus, they solely consider human problems in language learning when approaching a text, such as text length or overly complex syntax. Although these aspects could map to a higher difficulty for an automatic translation process, they only consider the problem in the source text as a comprehension problem, whereas in real-world scenarios most of the attention is on the target text, focusing on the essential cross-language aspects of terminology and pragmatics of the target language.

This dissertation represents an attempt at approaching this problem by transferring the knowledge from established MT evaluation metrics to a new model able to predict MT quality from the source text alone. To open the door to experiments in this regard, we explore the fine-tuning of a state-of-the-art transformer model (XLM-RoBERTa), construing the problem both as single-task and multi-task. Results for this methodology are promising, with both model types seemingly able to successfully approximate well-established MT evaluation and quality estimation metrics, achieving low RMSE values in the $[0.1 - 0.2]$ range.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2021 ELIS, the major European group dedicated to analyzing and reporting on the current developments of the language and translation sector, released its annual Language Industry Report stating that machine translation and post-editing represent the strongest future trend for both the industry and training institutions, especially those adhering to the European Master's in Translation (EMT) program.[1]

It is evident that machine translation has firmly established in the industry and research on this topic is thriving. Although quality improvements over the last few years have indeed been significant, the translation world has expressed a need, time and time again, for new methods and technologies to properly assess its quality, leading to the creation its own subfield: Machine Translation Evaluation. Most of the work in this regard has been focused on the resulting translation, both in traditional evaluation, where the machine translated segment is compared to a human reference translation, to the more recent quality estimation techniques, where the machine translated segment is evaluated on its own without any reference. The present work contributes to this field by seeking a different perspective, where the evaluation is centered around the source text. I define this as Machine Translation Suitability, formulating the following hypothesis:

> ***Hypothesis:*** *The better the machine translated version of a text, in terms of a well-established evaluation metric, the more suitable is the source text to machine translation.*

[1] As reported in the *2021 Language Industry Survey Report*. This URL was last consulted on 18/01/2022.

Starting from this hypothesis, we formulate the following research questions:

> ***Research Question 1:*** *Is it possible to accurately predict the MT Evaluation or Quality Estimation score from the source text alone?*
>
> ***Research Question 2:*** *If the first research question is true, is it better to construe the problem as a single-task or as a multi-task problem?*

Therefore, the objective of this dissertation is to create a supervised model for the prediction of machine translation quality from the source text alone (here defined as Machine Translation Suitability). In order to achieve this objective, we begin by reviewing the literature on machine translation evaluation. On the basis of this review, we compile an ad-hoc corpus pairing source text segments with the evaluation score of their automatic translations. Finally, the experiments are conducted using the state-of-the-art transformer model XLM-RoBERTa in two different settings: single-task and multi-task. This work offers a description of all the aforementioned steps and is articulated in the following chapters:

Chapter 2 provides a review of the key concepts that represent the core of this dissertation. It will initially provide an explanation of the field of natural language processing, explaining the concept of automatic text classification and introducing transformer models. A brief history of machine translation is provided before delving in the literature on machine translation evaluation and quality estimation, which represent the driving topics of the entire work. The Chapter closes with a review of the related work with respect to the current thesis.

Chapter 3 establishes the settings for the experiments and the research questions, as well as providing an overview of both the selected corpora and their preprocessing in order to construct the corpus on which to train the machine learning algorithm. It also provides an overview of the architectures for our models, explaining the structure of the transformer of choice (XLM-RoBERTa) and the fine-tuning techniques employed to obtain the predictions, either construing the problem as single-task or multi-task.

Chapter 4 presents the results obtained by the architectures fine-tuned on the corpus examined in the previous chapter. It discusses the technical and methodological limitations of the methods employed in this dissertation, while also offering considerations on possible practical applications and future research that could lead to validation and improvement of the proposed models.

Chapter 5 summarizes the dissertation and draws the final conclusions, reflecting on the research outcomes and paving the way for the following work.

Chapter 2

Background

2.1 Introduction

This Chapter aims to provide an overview of the essential building blocks that lie at the basis of this dissertation.

Section 2.2 provides a general definition of natural language processing (NLP), discussing the approaches which are salient for this work and the state-of-the-art (SOTA) models at the time of writing, focusing in particular on transformer models (Section 2.2.2).

Section 2.3 delineates a brief history of machine translation (MT), starting from rule-based models and ending with the contemporary data-driven approaches, with special attention to the current state-of-the-art, namely neural machine translation (NMT). Machine translation evaluation, both in its traditional form and more recently as quality estimation (QE), will be discussed, offering an overview of the most used metrics in research and of the latest published results in two of the most prominent conferences on machine translation, namely WMT2020 and WMT2021.

Finally, an overview of related work will be provided in Section 2.4, focusing on the PreDicT project by the University of Ghent, the APE-QUEST project founded by the European Union and the SmartLQA project proposed by Welocalize, one of the major language service providers in the world.

2.2 Natural Language Processing

Computer programming languages possess the characteristic of being formal-born languages, abiding to specific syntactic rules which completely avoid ambiguous statements by restricting their expressiveness and being usually based on context-free grammars, according to the classification first introduced by Chomsky (1959). Hence, they are inherently different from natural languages, because they instead possess fundamentally opposed characteristics, such as arbitrariness and discreteness, which are essential for human communication (Hockett and Hockett, 1960).

Natural language processing, or NLP, positions itself at the intersection between computers and humans and can be defined as follows:

> *Natural language processing is an area of research in computer science and artificial intelligence (AI) concerned with processing natural languages such as English or Mandarin. This processing generally involves translating natural language into data (numbers) that a computer can use to learn about the world.* (Hapke et al., 2019, p.4)

Natural Language Processing normally follows either one of two main approaches, namely rule-based NLP and statistical NLP. *Rule-based NLP* is defined by models based on hand-crafted rules or grammars which, combined with the data, allow the machine to obtain an "understanding" of Natural Language or — to be more precise — to obtain the desired output (Hapke et al., 2019, p.4).

Statistical NLP instead seeks to build models which are tuned on either unstructured or annotated data, without solely relying on rule-based methods. It draws heavily from the field of machine learning, employing algorithms to learn (hidden) patterns within the data and subsequently using these patterns to classify or predict an event related to the problem at hand (Alloghani et al., 2020). These algorithms can be further divided into *supervised* and *unsupervised*, depending on whether the training data includes labels for each data point or not (Alloghani et al., 2020).

Since this work will employ a supervised model, we will only focus on this paradigm in the following paragraphs.

2.2.1 Supervised Models

In order to build a supervised machine learning model, three types of data need to be provided, namely the input to be classified, the features to be predicted and the correct values for the prediction. Every instance in the dataset must be represented using the same set of features, which may be *binary* (falls within two categories, e.g. `POSITIVE`— `NEGATIVE`) *categorical* (falls within more than two categories, e.g. `POSITIVE`—`NEGATIVE`—`NEUTRAL`) or *continuous* (falls within a definite range, e.g. $[0, 1]$) (Kotsiantis et al., 2007).

Supervised algorithms attempt to learn to predict or classify input data from the features which are assigned to them. This process happens thanks to the sheer amount of data the algorithm has at its disposal. By being able to compare multiple data points sharing the same features, the algorithm attempts to predict these features by recreating an internal representation of the instances it is trained upon. This representation is then used to predict which label would be assigned to a new instance, which has not yet been seen by the model. By counting and measuring the errors, based on whether the unseen instance has been correctly predicted/classified, it is possible to determine the accuracy of the model using the same features provided during training (Alloghani et al., 2020). Additionally, supervised models can be further divided into *classification algorithms* if the model is designed to predict binary or categorical features and *regression algorithms* if the model is designed to predict continuous features (Alloghani et al., 2020).

Over the years, many different types of algorithms have been developed, starting from simple architectures such as K-Nearest Neighbours (kNN) (Cover and Hart, 1967), up to more complex mathematical models such as Support Vector Machines (SVMs) (Joachims, 1999) and the Perceptron (Kotsiantis et al., 2007).

This last model is the real core of the most advanced NLP methodologies and can be briefly described as follows: Given a data point and its features, a set of *weights* (W) is assigned to each of the features (X). The feature vector is thus denoted as:

$$X = [x_0, x_1, ..., x_i, ..., x_n]$$

with x_i representing a reference integer.

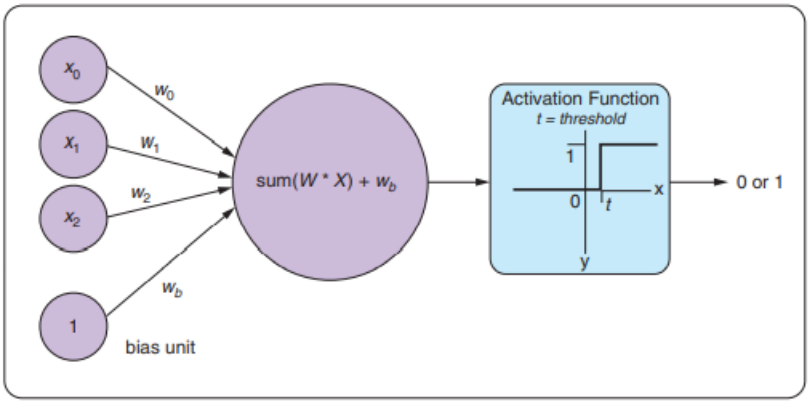

Figure 2.1: A graphical representation of the Perceptron (borrowed from Hapke et al. 2019:157)

Similarly, the weights vector will be denoted as:

$$W = [w_0, w_1, ..., w_i, ..., w_n]$$

where w_i corresponds to the index of feature x associated with that weight. The Perceptron then computes the sum of all weighted inputs

$$\sum_i x_i * w_i$$

and the output goes through an adjustable threshold, called *activation function* which determines whether the Perceptron will fire and output a 1, or not and output a 0 (Kotsiantis et al. 2007; Hapke et al. 2019, pp. 158).

The Perceptron shown in Figure 2.1 is in fact the basic unit composing artificial neural networks, large interconnected collections of perceptrons which currently represent the state-of-the-art method for most NLP tasks (Hapke et al., 2019).

2.2.2 State of the Art: Transformer Models

Although the paper "Attention is all you need" was limited to the field of machine translation, the world of Natural Language Processing has experienced

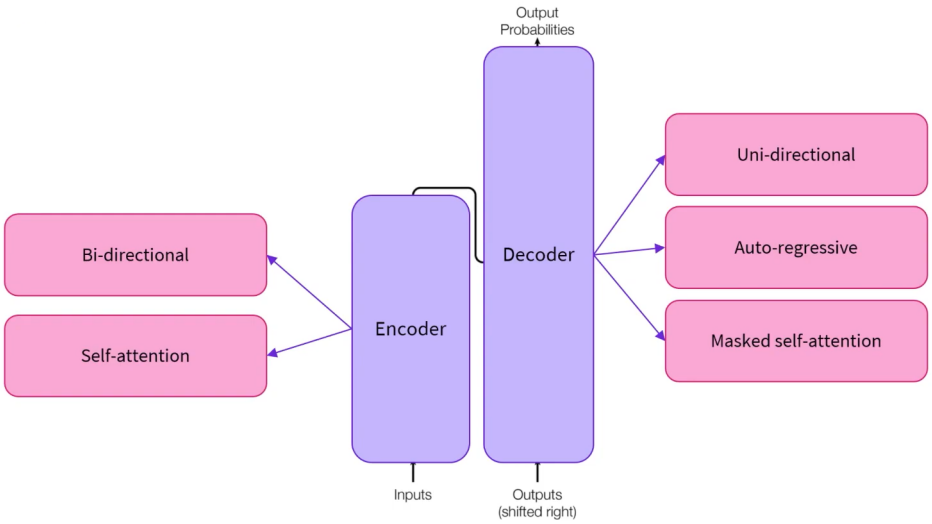

Figure 2.2: The transformer Encoder-Decoder architecture proposed by Vaswani et al. (2017) and its main characteristics (borrowed from the Huggingface course – `https://huggingface.co/course/chapter1/1`. Last consulted on 28/01/2022).

major improvements by using the same architecture proposed by Vaswani et al. (2017). It belongs to the family of artificial neural network architectures (ANNs), together with convolutional neural networks (CNNs) (LeCun et al., 1999), recurrent neural networks (RNNs) (Schuster and Paliwal, 1997) and Long Short-Term Memories (LSTMs) (Hochreiter and Schmidhuber, 1997), which had already proven to obtain very good results in a multitude of NLP tasks (Schmidhuber, 2015). But the architecture they developed significantly outperformed most other techniques across almost all fields of NLP and machine learning after the release of the first and most successful language model based on the transformer: BERT (Devlin et al., 2018). Further studies and its prominence across the field even spurred the creation of a field of its own, called BERTology (Rogers et al., 2020).

Since the transformer architecture is central to the scope of this work, in the following we will briefly introduce its structure and its various declinations. Figure 2.2 offers a general outlook of the structure of a transformer, as well as the main characteristics of its components.

2.2.2.1 Encoder models

The first type of transformer models are Encoders, also called autoencoding models. They correspond to the original encoding architecture described by Vaswani et al. (2017) and are represented by the left-most component in Figure 2.2. They are normally pretrained by corrupting input sentences, namely hiding one or more tokens in the input sentence. These are first tokenized without using a mask so that they are able to account for all the tokens simultaneously.

The output of an encoder, called a *feature vector*, is a numerical representation of the input sentence. Each token passed through the Encoder is assigned a numerical representation with a predefined length. These representations take into account both the context on the left and on the right of the token, since one of the main characteristics of Encoders is their *bidirectionality*, namely that they are able to process text not only left-to-right but also right-to-left at the same time (Hapke et al., 2019, p.311-312). Everything is then bound together thanks to the *self-attention mechanism*, which functionally relates the token to other tokens within the same sequence, selecting those that represent the meaningful context to be considered for the input (Hapke et al., 2019, p.313-316). These models are proficient at Natural Language Understanding tasks, such as Masked Language Modeling, where the objective is to guess a random hidden word within a sentence, or sequence classification/regression tasks (Hapke et al., 2019, p.317).

Examples of models used for these type of tasks are *BERT* (Devlin et al., 2018) and *XLM-RoBERTA* (Conneau et al., 2019).

2.2.2.2 Decoder models

The second type of transformer models are Decoders, also called autoregressive models. They correspond to the original decoding architecture described by Vaswani et al. (2017) and are represented by the right-most component in Figure 2.2. They are normally pretrained as language models, which means they read an input sentence word-for-word and have to guess the following token by considering only the previous ones. A so-called *mask* is applied on the full sentence to let the attention heads only see what came before in the text.

Although the decoder, like the encoder, also outputs a numerical repre-

sentation from the input sentence, also called a *feature vector*, it uses *masked self-attention*, which forces the model to be *unidirectional* by only feeding the model the context preceding the current token and not the one following it. Hence, these models reuse the output of the previous time step, namely the token previously predicted for the same sentence, and use it as additional input to "remember" past context (Hapke et al., 2019, p.311-117). These models are proficient at Natural Language Generation tasks, such as Causal Language Modeling, where the objective is to guess the next word in a sentence given the previous text, meaning it is optimal as a text autocomplete tool or for generating summaries and headlines of long articles.

Examples of models used for these type of tasks are are the GPT series, with the latest release being *GPT-3* (Radford et al., 2018, 2019; Brown et al., 2020).

2.2.2.3 Encoder-Decoder models

Encoder-Decoder models, also called sequence-to-sequence models (seq-to-seq), correspond to the full architecture in Figure 2.2. As seen in Section 2.2.2.1, the input text is first transformed into a *feature vector* by the Encoder, which then passes both this vector and a placeholder token to the Decoder. This placeholder serves as a prompt for the Decoder to begin the prediction, which will be based on the feature vector generated by the Encoder. The Decoder will thus proceed with its predictions autoregressively over every single word it outputs, generating text which retains the meaning contained within the *feature vector*.

Additionally, the length of the output sequence is independent from the one in the input, thus making these models optimal for tasks such as Text Summarization, Question Answering and, most notably, Machine Translation (Hapke et al., 2019, p.311-317).

2.3 Machine Translation

Across all the fields of natural language processing, the field of machine translation is arguably one of the most successful and popular. Indeed, over the last decade, the translation industry has experienced a revolution thanks to the increasing implementation of machine translation in both industrial and institutional settings, with the 2021 Language Industry Report stating that

machine translation and post-editing represent the strongest future trend for all respondents[1] Some experts even claimed that it will soon replace human translators (Wu et al., 2016), but in reality its usage has been shown to be mostly beneficial to professional translators, increasing their productivity and becoming one of the sharpest tools in their toolkit (Koehn, 2020).

With the aim of contextualizing the research on MT, this Section will provide a brief history of machine translation, from the early rule-based architectures to the state-of-the-art transformer models. Given the specific focus of the work, special attention will be given to the methodologies developed for machine translation evaluation, both in their traditional form and in the more recent quality estimation variant.

2.3.1 Rule-based MT

The earliest attempts towards the development of machine translation systems all employed techniques based on monolingual and bilingual dictionaries and hard-coded rules regarding morpho-syntactic, lexical and generational features (Quah, 2006). This meant that many such features had to be derived directly from formal grammars and translated manually into algorithms which could be interpreted by the computers. The amount of human intervention required was enormous and between the 1950s and 1980s progress was relatively slow (Okpor, 2014).

It is possible to distinguish between two main approaches that characterized Rule-Based Machine Translation (RBMT), as exemplified by the Vauquois triangle, displayed in Figure 2.3:

1. the Direct approach

2. the Indirect approach (either Interlingua or Transfer)

The *Direct Approach* characterizes the very first attempts at MT, where it was initially thought possible to perform a word-for-word substitution between the source language and the target language, completely disregarding any linguistic analysis (Quah, 2006). Since they are specifically designed for

[1] As reported in the *2021 Language Industry Survey Report*. This URL was consulted on 18/01/2022.

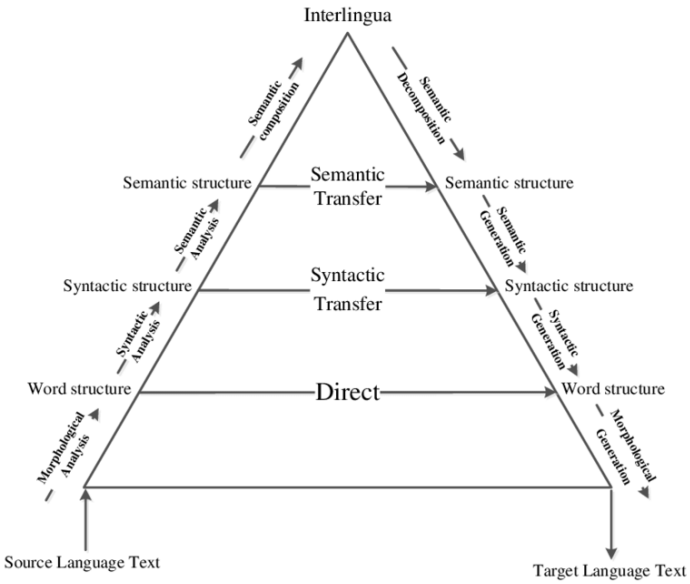

Figure 2.3: The Vauquois Triangle (1968). The left side represents the analysis step, while the right side represents the generation step.

not only one single language pair, but also one single language direction, these early attempts have immediately shown their inadequacy for the task at hand, with models that were not only rigid but also unable to handle idiomatic expressions, ambiguity and language pairs which did not share the same sentence structure (Quah, 2006; Okpor, 2014).

The *Indirect Approach*, in both its realizations as **interlingua** or **transfer**, aims at including both morpho-syntactic and semantic features via hard-coded rules. Following Hutchins and Somers (1992), the two methodologies can be defined as follows:

Interlingua refers to the process of translating the text in two stages. Initially, the source text is converted into a semantic representation, supposedly language-independent, and then translated into the target language(s) based on dictionaries and formal grammars.

Transfer refers to the process of generating two separate representations for the source text and the target text, with the system first generating a representation of the source text (`analysis`) which is then used to produce a

representation of the target text (`transfer`) to be ultimately converted into the target text (`generation`).

2.3.2 Corpus-based MT

The end of the 1980s saw the beginning of a new era for machine translation, which began to incorporate ideas from a newly emerging and rapidly growing field: corpus linguistics. In particular, researchers started to employ new approaches based on parallel corpora, which soon came to replace the rule-based architectures dominating the field in the years prior (Koehn, 2009, p.17). Machine translation methods which avoid using linguistic information and are entirely based on algorithms for the inference and extraction of parallel segments from extremely large aligned corpora are called either *corpus-based* or *data-driven* (Quah, 2006; Okpor, 2014). This approach has proven to be so effective that it is still at the core of contemporary state-of-the-art architectures and has developed in two major model types (Wang et al., 2021): Statistical Machine Translation and Neural Machine Translation

2.3.2.1 Statistical MT

The concept behind Statistical Machine Translation (SMT) was first described by Brown et al. (1990), following the development by IBM of the first system of this type in 1989. In its most basic form, it defines the idea of automatically learning translation knowledge by applying statistical methods over large amounts of bilingual textual data instead of relying on hard-coded rules. The objective of these systems is learning to translate by analysing the statistical relationships between the source text and its human target translation. Systems following this method feature three main components, namely the Translation Model, the Language Model and the Decoder.

A *Translation Model* is a statistical model trained on parallel bilingual corpora. It seeks to retrieve the target segment with the highest probability score among all aligned segments, conditioning the search based on the input text.

A *Language Model* is a statistical model trained on a monolingual corpus. Its objective is also to produce the segment with the highest probability score, but without considering the source text.

As (Specia, 2010, p.III) aptly states:

> *These two components are usually seen as proxies to what is considered to constitute a good translation, respectively: adequacy and fluency.*

The final component is the *Decoder*, which searches for the best possible translation within the space of all possible translations based on the probability estimates of the Language Model and the Translation Model (Specia, 2010).

Although RBMT approaches remained valid until the late 2000s, Statistical Machine Translation gradually superseded them with staggering improvements thanks to the development of translation units considering spans smaller than the whole sentence. At first, these approaches were exemplified by the *word-based* method, where a parallel bilingual corpus aligned at the sentence level is employed to create a so-called translation table (**t-table**) from word alignments. This table shows, for each given word, all its possible translations with the corresponding probability estimates, with the assumption of a direct word-for-word correspondence between the translations (Koehn, 2009).

While this method showed considerable promises, it was only with *phrase-based models (PBMT)* that SMT managed to showcase its potential. PBMT models inherit the same approach as the word-based models, but instead of working at the word level, they construct the t-table using n-grams — contiguous sequences of n tokens — usually with $n = 3$. By considering a span of three words, these models were able to better account for contextual information and collocations, thus showing significant improvements in terms of fluency as well as word insertions. One of their major shortcomings, though, was their inability to model long-distance dependencies, leading to problems when attempting to reorder long sentences effectively (Koehn et al., 2003).

In an attempt to overcome these limitations, *Syntax-based systems* which included dependency parsing tags on both the source and target text were developed, showing promising initial results. Nonetheless, these were still limited by the increased memory requirements, leading to a significantly slower performance and excessive reliance on scarcely available dependency parsers (Williams et al., 2016).

2.3.2.2 Neural MT

As had previously happened with SMT, a new turning point came for machine translation in 2015, when development in the field of machine learning led to the creation of a new SOTA: Neural Machine Translation systems (Bojar et al., 2016). Although the idea of applying neural networks to machine translation had been advanced before, their implementation had always posed considerable challenges due to the sheer amount of data required to obtain acceptable results, in conjunction with their considerable computational cost (Koehn et al., 2020).

NMT models are usually constructed as sequence-to-sequence or *seq-to-seq* models, because they take a sequence as input (the *source text*) and predict another sequence as output (the *Hypothesis*) (Bahdanau et al., 2014). This process happens often under the hood, using training data to compute a vector space and assigning a single vectorial representation to each token, called a word embedding. Each word thus obtains a mathematical representation which captures both semantic and morpho-syntactic properties and onto which similarity scores can be computed, enabling models to cluster semantically related words together. This process is achieved using the Encoder-Decoder structure described in Section 2.2.2.3, hence they are also often referred to as *seq-to-seq* architectures.

In their first form, the preferred neural network types for this task were Recurrent Neural Networks (RNN) given their ability to retain contextual information from the output at earlier steps, which is crucial for sequential tasks such as machine translation, because it allows the system to account for previous words in the input segment when passing from the Encoder to the Decoder (Bahdanau et al., 2014). Since RNNs had to include all previous information in their output, they were also prone to errors due to noise caused by long distance dependencies, which were often not maintained. Hence, they were soon replaced by another form of RNNs, namely Long-Short Term Memories (LSTM), which are able to better handle the dependency problem by selecting only useful information to carry over to the following step thanks to their so-called *forget gate* (Sutskever et al., 2014).

The major breakthrough, though, came with the implementation of the attention mechanism to NMT models (Vaswani et al., 2017). An additional neural network was added to the previous architecture, called a *self-attention layer*, which is trained in parallel to the rest of the model. Every time a new

token is provided to the feedforward neural network, rather than building a single context vector out of the last encoder hidden state, this separate layer computes its relationship with all other tokens in the segment and selects which information is relevant for the current context vector.

Essentially, by supporting the decoder with information regarding the relationship among tokens during training, this methodology significantly improved output quality, especially with regards to long-term dependencies, as well as model training time, since it does not employ any recursive mechanism and vectors flow through the stack simultaneously instead of processing one token at a time. Thanks to these improvements, this architecture has established itself as the state-of-the-art architecture for machine translation to date (Tan et al., 2020).

This new paradigm brought considerable attention to the MT field, with researchers even claiming that machine translation would soon be on par with human translators (Wu et al., 2016). Nonetheless, in reality numerous studies have disputed this view, stating that while these architectures have significantly improved in quality and accessibility, they still suffer from underlying problems which are yet to be solved (Bentivogli et al., 2016, 2018). Hence, plenty of work has also been devoted to the evaluation of machine translation systems, utilizing a plethora of different techniques which will be addressed in the following section.

2.3.3 Machine Translation Evaluation

Ever since the very first implementation of machine Translation systems, both researchers and the industry have had to face the challenge of evaluating a translation offered by these systems. Quality is evaluated either manually by professional translators or automatically with dedicated metrics which compare the system output (*hypothesis*) against one or more translations provided by humans (*reference*) (Koehn, 2009).

Manual evaluation is usually considered the golden standard for the proper evaluation of MT outputs, as it ensures that complex linguistic phenomena are recognized and specific error typologies, such as MQM, are respected (Lommel et al., 2014). Its major drawbacks lie in the fact that it is inherently slow and expensive, while also being quite difficult to reproduce consistently as it needs to be confirmed by inter-annotator agreement measures to ensure

the objectivity of the evaluation (Castilho et al., 2018). Automatic evaluation, instead, provides an easily reproducible, fast tool to assess the quality of a system while simultaneously allowing for cross-system comparison. Major drawbacks include its inability to accurately indicate error severity, its limited capacity of accounting for syntactic or semantic equivalence and their intrinsic bias towards only one or at most a few of the multiple possible translations of a source sentence (Castilho et al., 2018; Kocmi et al., 2021).

Although the field of Machine Translation Evaluation has produced a plethora of metrics (Marie et al., 2021), for the scope of this work we will only be focusing on automatic metrics. Thus, in the following we will introduce three among the most prominent metrics used by the research community according to the latest WMT metrics shared tasks (Freitag et al., 2021): BLEU, variations of LEPOR and BERTScore.

2.3.3.1 BLEU

The standard among all automatic translation quality assessment metrics is certainly BLEU, with 98.8% of MT papers using it as their primary metric for comparing results with other systems (Marie et al., 2021). It is a rule-based metric which computes a similarity score between the hypothesis and a reference human translation (Papineni et al., 2002). This similarity score is computed by matching the respective n-grams of the two segments within a [1,4] range, while also including a penalty factor for overly short translations (Papineni et al., 2002). Being a similarity metric, its score ranges between 0 and 1, with 0 meaning there is no similarity between the hypothesis and the reference and 1 meaning that the two segments are exactly identical.

Since it has shown an erratic behaviour when modifying its hyper-parameters, an updated version called sacreBLEU was introduced to offer a standardized implementation for the research community with precise indications regarding the pre-processing steps to be carried out to ensure comparability across systems (Post, 2018).

Nevertheless, at its core, it remains a tool which simply compares two sentences based on their n-grams, meaning that it has significant limitations with respect to correctly assessing the suitability of a translation to the tokens actually present in the segment, thus remaining highly susceptible to the

high degree of variability of inflection or word order choices. Additionally, it has been consistently shown that the state-of-the-art metrics now greatly outperform its correlation with human judgments and several other metrics and methods have been suggested. Nevertheless, it is still being largely employed in the community, especially as a baseline metric, due to its ease of use and easily understandable scoring system. (Reiter, 2018; Mathur et al., 2020).

2.3.3.2 hLEPOR and cushLEPOR

Another n-gram based metric is LEPOR, which was designed as a combination of word order penalty and precision, recall, and an enhancement of sentence-length penalty (Han et al., 2012).

The hLEPOR variant, using the harmonic mean to group factors and calculate the final score, achieved the best performance on the English-to-other system level evaluation task in ACL-WMT13 (Macháček and Bojar, 2013). It features a set of tunable parameters, which in the original implementation were selected empirically and set as default.

Below is a brief description of each parameter, as explained by Han et al. (2021):

1. **alpha:** the tunable weight for recall
2. **beta:** the tunable weight for precision
3. **n:** word count before and after matched word
4. **weight_elp:** tunable weight of enhanced length penalty
5. **weight_pos:** tunable weight of n-gram position difference penalty
6. **weight_pr:** tunable weight of harmonic mean of precision and recall

In this recent publication, they also released cushLEPOR, a customised version of hLEPOR built via automatic tuning of the aforementioned weighting parameters using pre-trained language models. Additionally, for the language pairs English-German and Chinese-English, it was optimised towards a human gold standard based on the MQM and pSQM frameworks, as required by the WMT2021 task (Han et al., 2021).

Despite the major drawback of having to perform the optimization for each language pair independently, this methodology has shown a significant boost in the performance of hLEPOR in terms of correlation with human judgments, further increasing the gap with BLEU and establishing cushLEPOR as one of the most advanced state-of-the-art metrics for traditional n-grams-based translation quality assessment tasks (Han et al., 2021; Freitag et al., 2021).

2.3.3.3 BERTScore

BERTScore belongs to the family of metrics which employ embedding similarity as their preferred method to score sentence similarity. Analogously to the previous metrics, it computes a similarity score between the hypothesis and the reference translation segment at the token level. In this case, though, the computation is not based on exact matches between n-grams but instead leverages contextual embeddings (Zhang* et al., 2020).

Each token within the two sentences to be compared is assigned their respective representation based on BERT, which considers different vector representations for the same word depending on their surrounding words. Pairwise cosine similarity is then computed among each token in the two segments. Finally, the reference and candidate tokens are matched greedily based on these scores (Devlin et al., 2018).

BERTScore has been shown to highly correlate with human judgment on sentence-level and system-level evaluation and it is one of the baseline metrics selected for the WMT2021 Metrics Shared Task (Freitag et al., 2021).

2.3.4 Machine Translation Quality Estimation

Machine Translation Quality Estimation (QE) has the objective of predicting Machine Translation quality automatically without looking at a reference translation (Specia and Shah, 2018). The reason behind this lies in the considerable costs behind obtaining large enough quantities of reference translations. It can be employed to select the best translation when several translation engines are available or can inform the end user about the reliability of automatically translated content. Many recent MTQE models employ multilingual pre-trained representations from very large language

models, which have led to impressive results in the past few years (Specia et al., 2020).

Although many of these models are indeed difficult to make portable due to their size, research in this field has been steadily increasing and new architectures and methods are being proposed on a yearly basis, especially thanks to the specific task developed by WMT for this field of study. It is also important to note that, contrary to traditional metrics, in this case the predictions are often not bound to the $[0, 1]$ range, thus hindering their transparency and ease of use. Additionally, one recent paper has disputed certain approaches to QE, suggesting that although quality estimation models might capture fluency in the *hypothesis* and complexity of the *source text*, they might not be able to model adequacy of translations effectively (Sun et al., 2020). At present, it appears that no single metric is being consistently deployed to production in the industry or institutions, with the top systems from WMT2020 being outperformed by more recent submissions (Specia et al., 2021). Nonetheless, experiments towards the development of a QE pipeline for the E-Translation platform have been subsidized by the European Union in the form of the APE-QUEST project, which seeks to employ QE methods to reduce costs and time for the translation pipeline by automatically rerouting source texts to either a human translator, a machine translation system or to an automatic post-editing component. This project will be thoroughly explained in Section 2.4.

In this Section we will be going over two of the best-performing frameworks for machine translation quality evaluation, namely COMET and TransQuest.

2.3.4.1 COMET

COMET is a PyTorch-based framework for training multilingual MT evaluation models that can function as metrics, built on top of another QE framework by Unbabel called OpenKiwi. It generates prediction estimates of human judgments in the form of Direct Assessments (DA) (Graham et al., 2013), which fall within a $[0, 100]$ range, Human-mediated Translation Edit Rate (HTER) (Snover et al., 2006) and metrics compliant with the Multidimensional Quality Metric framework (Lommel et al., 2014; Rei et al., 2020).

COMET features both a traditional reference-based architecture as well

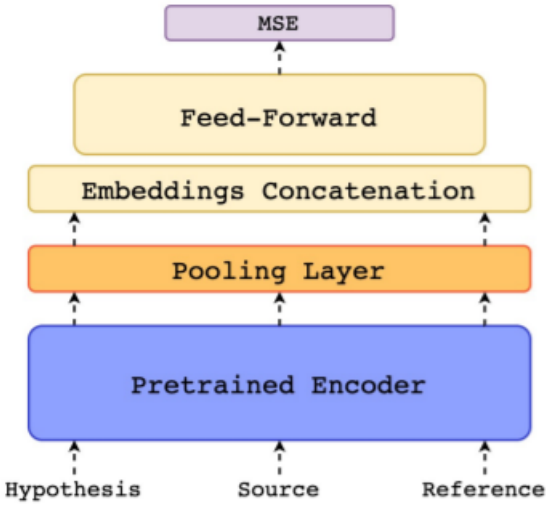

Figure 2.4: COMET model architecture (borrowed from Rei et al. 2020

as a reference-less based one, but for the scope of this work we will be focusing on `wmt-large-qe-estimator-1719`, which is the reference-less quality estimation component. Similarly to BERTScore, at their core, all the models within this framework leverage a pretrained, cross-lingual model, namely XLM-RoBERTa (base) as their Encoder.
Starting from the input sequence:

$$X = [x_0, x_1, ..., x_n]$$

the Encoder outputs an embedding $e_j^{(l)}$ for every single token x_j and layer $l \in \{0, 1, ..., k\}$. Afterwards, these vectors are passed through a pooling layer to create sentence embeddings at the segment level. These serve as input to a feed-forward regressor, as displayed in Figure 2.4. The training objective of the model is to minimize the Mean Squared Error (MSE) between the predicted scores and the Quality Assessments, either DA, HTER or MQM (Rei et al., 2020).

This architecture has performed remarkably well at WMT2021 (Freitag et al., 2021) and both the reference-based DA and the reference-less QE model were evaluated as the best performing metrics in a large-scale study performed by Microsoft Research (Kocmi et al., 2021).

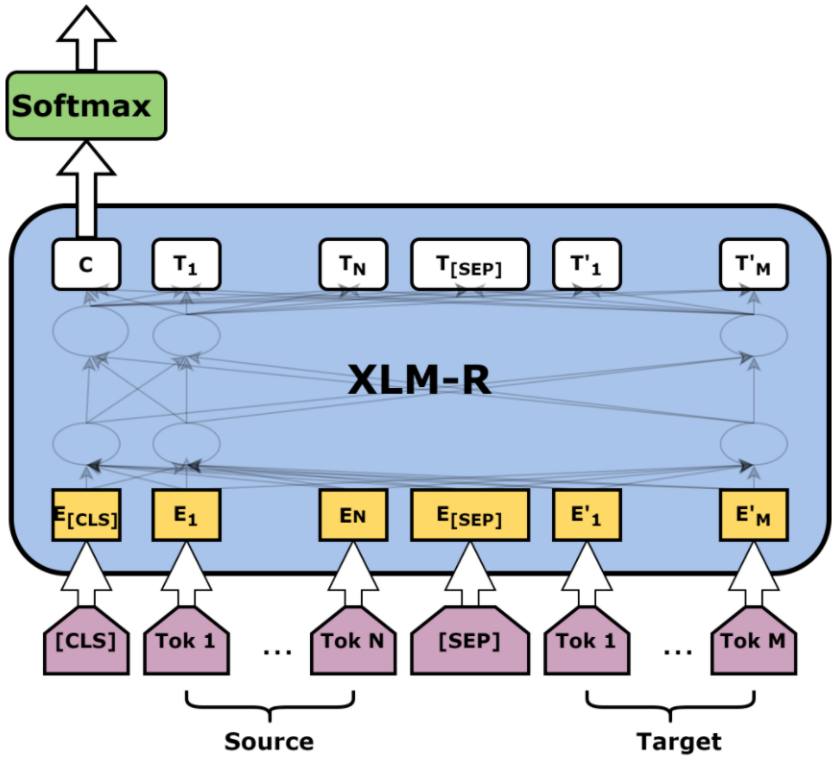

Figure 2.5: MonoTransQuest model architecture (borrowed from Ranasinghe et al. 2020a)

2.3.4.2 TransQuest

TransQuest is an alternative to the OpenKiwi framework for training open-source quality estimation models. It offers two distinct architectures, one of which will be employed in this work as it consistently outperformed the others, namely `MonoTransQuest` (Ranasinghe et al., 2020a,b).

The `MonoTransQuest` architecture proposed by Ranasinghe et al. (2020) for sentence-level classification takes a sequence of tokens as input, separated by a `[CLS]` token, and the source and hypothesis sentence tokens, separated by a `[SEP]` token (see Figure 2.5). The entire string is subsequently fed to a transformer encoder to obtain a representation on which a cross-entropy classification head is adopted as the loss function. Similarly to COMET, the language model selected as the Encoder was XLM-RoBERTa.

Several language combinations were released, as well as models supporting any-to-any and any-to-en language pairs. All models are domain independent (Ranasinghe et al., 2020a).

This model obtained the top level correlation score for sentence Direct Assessment in WMT2020 (Specia et al., 2020) and has been used as one of the baseline systems in WMT2021, as well as a "teacher" for a smaller model — hence more easily portable and reproducible — using knowledge distillation techniques, by the BERGAMOT group who participated to the same task (Freitag et al., 2021).

2.4 Related Work

PreDicT is an ongoing research project by Ghent University to develop a "translatability prediction system" for the English-Dutch language pair that not only assigns a global difficulty score to a source text, but also identifies which passages are more problematic for translation.

They employ the term *translatability* defined as "the difficulty of a translation task", in opposition to *readability* as the "difficulty of a monolingual text" and argue that, although the two might overlap in some regards, a translation task cannot be solely defined based on monolingual features (Vanroy et al., 2019). They instead hypotesized three difficulty predicting features: number of errors in the final target text, word translation entropy (i.e. the number of different translation options) and syntactic equivalence between the source and target text. They found that these features correlate with translation process data obtained from keystroke logging and eye-tracking during the translation process. Hence, they propose to treat these features as proxies for cognitive effort and use them as predictors for translatability. Albeit promising, this work solely addressed human-translation difficulty and no study tailored to MT systems has yet been published.

The Automated Post-editing and Quality Estimation project (APE-QUEST) was an experimental project which ran from 2018 to 2020 in conjunction with the European Union, the University of Sheffield and Crosslang (Depraetere et al., 2020). It proposed what they called a *Quality Gate* for the automatic quality estimation of domain-specific machine translated segments. The objective of this gate was to obtain an acceptable translation quality by automatically rerouting a translated segment to either a human post-editor

or to a QE component for automatic correction. They define translation acceptability based on two different use cases: *assimilation*, allowing a reader to understand the gist of a text through its translation, and *dissemination*, meaning external use. The threshold for accepting or rejecting a machine translated segment is set based on their relative QE score. In their work, they tested both with a QE value < 0.9 and < 0.8, comparing the results on time and quality gains against a machine translation centered workflow (Vanallemeersch and Szoc, 2020).

The group has trained both QE and Automatic Post Editing (APE) systems for three language pairs (`en>pt`, `en>nl` and `en>pt`) on texts mostly relating to the legal domain, for which they also released a corpus (Ive et al., 2020). The evaluation showed that the APE component did not seem to produce satisfactory results because of the already generally high-quality output from the MT system they employed, making it difficult to produce exhaustive automatic corrections. The QE component instead showed good potential, since it consistently improved cost and time measures in both assimilation and dissemination cases, without strongly compromising translation quality with respect to a traditional human-only workflow, especially in the dissemination use case. They thus concluded by underlining the potential of the Quality Gate, especially as concerns the quality estimation component (Vanallemeersch and Szoc, 2020).

One last project related to this dissertation is the one carried out by Welocalize, one of the major Language Service Providers (LSP) worldwide. During the MTSummit2021, they presented SmartLQA (Smart Linguistic Quality Assessment), an ongoing project on the analysis of the impact of the source text on machine translation (Yanishevsky, 2021). Their work handled the prediction of 'at-risk' content prior to translation, analyzing the linguistic aspects within the source text which are more likely to cause mistranslations and omissions in machine translation and post-editing. They employ both linguistic features and readability tests, such as the Flesch–Kincaid metric, concluding that a poor source text quality leads to poor target text quality. They demonstrate this also thanks to significant reductions in both time and costs for linguistic quality assessment when employing their methods to production. Although no predictive model using these features has been released, their work lays the ground for further studies on the impact of the source on the target text and further motivates the study of machine translation suitability.

Chapter 3

Experimental Framework

3.1 Introduction

This Chapter explicates the methodology used in this work, offering an analysis of the both the data and learning algorithms used to obtain the results in Chapter 4.

Section 3.2 provides the main hypothesis at the basis of this dissertation as well as the two research questions that this work seeks to answer.

Section 3.3 delineates the steps taken to collect and build the corpora used for this dissertation. It also includes a Section dedicated to their statistical analysis in order to assess their validity for the task at hand.

Finally, Section 3.4 elaborates on the architecture and machine learning techniques used in this work, concluding with an explanation of the evaluation metric used to obtain the results in Chapter 4.

3.2 Experimental Settings

Although their purpose is to give insights on the model's performance, it should be possible to exploit the MT metrics discussed in Section 2.3.3 to infer the difficulty of the translation task. This follows the intuition that the performance of a model on a given segment indirectly indicates how problematic that segment was to translate for the MT system.

Hence, we formulate the following hypothesis:

> ***Hypothesis:*** *The better the machine translated version of a text, in terms of a well-established evaluation metric, the more suitable is the source text to machine translation.*

In order to test this hypothesis, we propose to produce a corpus of source text segments, annotated with the corrisponding evaluation score of their automatic translations in order to train a model to predict such a score from the source text alone, thereby mimicking the estimation of translation suitability. With such a model, it would be possible to know how well an MT engine would perform on that segment and thus how "suitable" it would be for an MT engine to translate. This would allow for the development of a tool which can determine whether a source segment should be translated by an MT engine alone, be flagged for post-editing or be directed to a human translator, in a similar fashion to the APE-QUEST pipeline (see Section 2.4). As a first step to test this hypothesis, in this work we aim to answer the following research questions:

> ***Research Question 1:*** *Is it possible to accurately predict the MT Evaluation or Quality Estimation score from the source text alone?*

> ***Research Question 2:*** *If the first research question is true, is it better to construe the problem as a single-task or as a multi-task problem?*

The proposed experimental setting to corroborate these research questions is the following:

1. Compile a parallel corpus covering multiple domains of investigation.
2. Translate the source segments with multiple translation engines to reduce bias towards a single system.
3. Select several automatic evaluation metrics and compute the evaluation for the MT against the reference translations in the parallel corpora.
4. Fine-tune and evaluate a transformer model on the scores obtained in (3) considering only the source text as input. The experiments will be conducted both with a separate model for each label and a multi-task model encompassing all 4 scores. The model selected for this work is XLM-RoBERTa (Conneau et al., 2019).

Subcorpus	Translation Units	avg char en	avg char de
Europarl	1 916 741	151.45 ± 90.54	170.47 ± 101.74
Ubuntu	285 721	137.48 ± 69.28	165.38 ± 84.08
News	13 117	33.20 ± 74.58	40.14 ± 88.73

Table 3.1: Statistics of the three original corpora extracted from OPUS (Europarl, Ubuntu, News).

3.3 Corpus Collection

In order to proceed with the construction of the corpus, the first step was to select a relevant language pair for the purposes of the study. In the literature, the `en>de` language pair is especially prominent for both MT Evaluation and quality estimation (Specia et al., 2020), thus we decided to build a corpus solely for this language pair (Freitag et al., 2021; Marie et al., 2021; Specia et al., 2021).

Due to time limitations, we opted to utilize corpora which have been extensively used in MT research, namely those belonging to the OPUS project (Tiedemann, 2012). OPUS is a growing collection of translated texts from the Web which aims to provide open-source parallel corpora to the research community. We adapted three main corpora:

Europarl is a parallel corpus which features the proceedings of the European Parliament from 1996 to 2012[1].

Ubuntu is a parallel corpus of localization files for Ubuntu[2].

News-commentary v16 is a parallel corpus of news commentaries published during WMT19 for SMT model training[3].

An overview of the statistics of these corpora is available in Table 3.1. Although these corpora have been already extensively used in the literature, their pre-processing is done automatically, without any type of manual corrections. To ensure their quality, two additional filtering steps have been carried out on the translation units (TUs), following the approach of the WMT2020 task on parallel corpus filtering (Koehn et al., 2020).

[1] `https://opus.nlpl.eu/Europarl.php`. Last consulted on 20/11/2021
[2] `https://opus.nlpl.eu/Ubuntu.php`. Last consulted on 20/11/2021
[3] `https://opus.nlpl.eu/News-Commentary.php`. Last consulted on 20/11/2021

The process was divided into two steps:

1. `Step 1`: removal of long and very short segments from the corpora. In particular, pairs with source or target segment length falling outside a threshold are discarded from the dataset. In our specific case, this threshold is set to a minimum length of 25 characters and a maximum length relative to each partition and language, since the Ubuntu corpus exhibits significant differences in terms of segment length. Hence, the maximum TU length is determined by summing the average length of the TUs in a corpus, relative to their language, and one unit of their standard deviation.

2. `Step 2`: adaption of the filtering approach implemented in the open-source version of ModernMT[4]. Translation units are discarded if either the source or target segment character length (including whitespaces) exceeds the length of the other segment by more than 50%. In order to prevent the filter from discarding short valid sentence pairs, an arbitrary value of 15 is added to the initial character count. This allowed to identify and discard misaligned segment pairs. Following is a definition of the function, where *max_len* is the longest segment and *min_len* the shortest segment in the TU:

$$\frac{(max\_len + 15)}{(min\_len + 15)} > 1.5$$

3.3.1 Segment translation and scoring

For the translation step, a randomized set of TUs was extracted from the pre-processed OPUS corpora and merged together to generate a new balanced corpus. The English segments contained in this corpus were then translated into German using two out-of-the-box NMT systems: ModernMT[5] and Microsoft Translator[6] via their dedicated APIs. Both of them are based on the state-of-the-art transformer architecture (see Section 2.2.2) and trained on a large pool of parallel data. Although ModernMT would offer an adaption mechanism to adapt on out-of-domain data on the fly (Bertoldi et al., 2018),

[4] `https://github.com/modernmt/DataCollection/blob/dev/baseline/filter_hunalign_bitext.py`. Last consulted on 01/02/2022.

[5] `https://www.modernmt.com/`. Last consulted on 01/02/2022.

[6] `https://translator.microsoft.com/`. Last consulted on 01/02/2022.

we only implement the baseline system as domain adaption is beyond the scope of this work. Microsoft Translator does not provide any adaption and is implemented in its baseline form.

The resulting translations were paired with their respective TUs, forming a triplet of `[source, reference, hypothesis]` for the evaluation step. After a thorough review of the literature (see Section 2.3.3), a set of four evaluation metrics were selected, namely:

1. hLEPOR[7] — Alpha = 2.95, Beta = 2.68, n = 2, weight_elp = 2.95, weight_pos = 11.29, weight_pr = 1.87[8]

2. BERTScore[9]

3. COMET[10] – Architecture: `wmt20-comet-qe-da`

4. TransQuest[11] — Architecture: `monotransquest-da-en_de-wiki`

Since there is no direct implementation of cushLEPOR, we have implemented hLEPOR with the suggested settings for the `en>de` language pair provided in the original release (Han et al., 2021). BERTScore follows the standard implementation provided on the official GitHub repository. In the case of COMET, although the newest release forces the score to be within a $[0, 1]$ range, making it more easily comparable with the traditional scores, our implementation utilizes the early release `wmt20-comet-qe-da`, which only provides an unbounded score, because it was the only one available while building this corpus. The last metric is TransQuest, of which we implement the `en>de` version `monotransquest-da-en_de-wiki` instead of the multilingual model because of its better performance, as reported by the authors (Ranasinghe et al., 2020a,b). Both versions are made available on GitHub and hosted over HuggingFace.

The single scores were subsequently stored in a vector following the order of the list above and combined with the previous triplet.

[7] `https://github.com/poethan/cushLEPOR`
[8] For a thorough explanation of the parameters, see Section 2.3.3.2
[9] `https://github.com/Tiiiger/bert_score`
[10] `https://github.com/Unbabel/COMET`
[11] `https://huggingface.co/TransQuest/monotransquest-da-en_de-wiki`

source	reference	hypothesis	scores
Mr President, my group is very aware of the responsibility it bears in this debate today.	Herr Präsident, meine Fraktion ist sich der ihr zukommenden Verantwortung in der heutigen Debatte sehr wohl bewußt	Herr Präsident! Meine Fraktion ist sich der Verantwortung, die sie heute in dieser Aussprache trägt, sehr bewusst.	[0.8924, 0.4872, 0.6796, 0.7212]
The video, posted four days before the election, was watched more than 400,000 times.	Das vier Tage vor dem Wahltermin gepostete Video wurde mehr als 400.000 aufgerufen.	Das Video, das vier Tage vor der Wahl veröffentlicht wurde, wurde mehr als 400.000 Mal angesehen.	[0.8702, 0.6117, 0.7224, 0.7427]
Is used to notify that the table column header has changed	Wird zur Benachrichtigung bei Änderungen an der Überschrift der Tabellenspalte verwendet	Wird verwendet, um zu benachrichtigen, dass sich die Tabellenspaltenüberschrift geändert hat	[0.5871, 0.4711, 0.4749, 0.7373]

Table 3.2: Instances extracted from the final corpus with their evaluation scores

Examples extracted from the complete corpus are provided in Table 3.2. From the corpora in Table 3.2 the actual training and test datasets for our experiments were generated, pairing the source segments with their respective scores by combining only the `source text` and the `scores`. Thus, we obtained two separate training and test datasets for each system, containing 14,253 and 1584 instances respectively.

In the following Section an in-depth analysis of the score distributions for the training set is provided.

3.3.2 Distribution analysis

Before moving on to the actual experiments we here propose an analysis of the distributions of the labels to be predicted. We first analyze the expected distribution of the scores for the COMET metric, as reported by the main

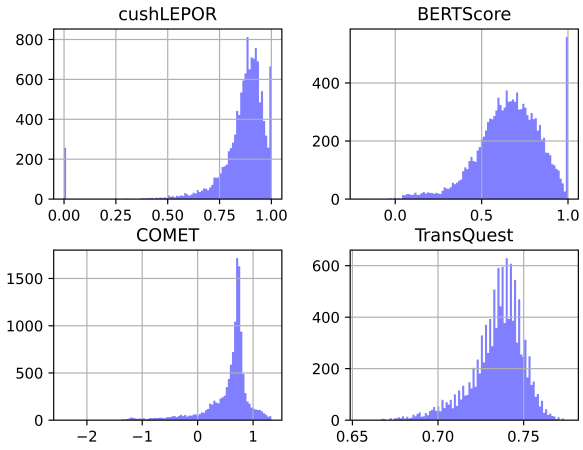

Figure 3.1: Distribution of the scores from the training corpus.

page of the library[12]. The median system score is 0.4159, meaning that systems with score < 0.3138 are in the bottom 25% and systems with score > 0.5828 are in the top 25%. Table 3.3 shows the description for the training set of ModernMT and indeed the system obtains quite a high performance, with a score of 0.5651. In an ideal scenario, where the parallel corpus had been manually compiled and thus surely contain never-before seen texts, this score could be simply accepted as such. In our case, though, the original corpora used for this work are open-source and specifically designed for NMT training (Tiedemann, 2012), meaning that it is highly likely that the underlying source texts have already been seen by the MT systems during training. For the scope of this research, this would be problematic because an attempt at learning machine translation suitability using these corpora would not be necessarily applicable to unseen texts.

Hence, we decided to compare the distributions of the training corpus to those of a new, smaller corpus, whose texts have not been seen by either systems. If the scores distribution of this secondary corpus were very similar to that of the training corpus, it would mean that either the TUs of the latter were never seen by the MT system or that there is no significant

[12] `https://unbabel.github.io/COMET/html/models.html`. Last consulted on 04/02/2022.

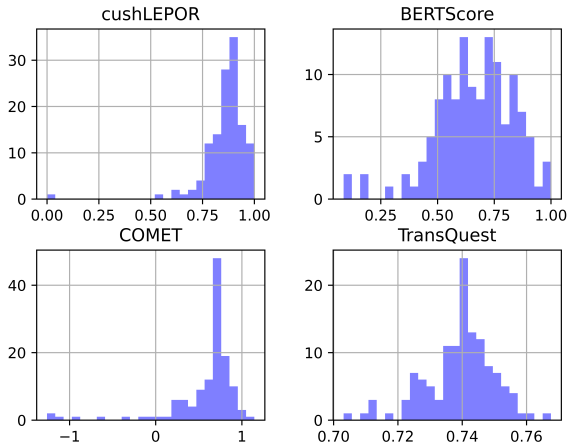

Figure 3.2: Distribution of the scores from the Globalvoices corpus

	cushLEPOR	BERTScore	COMET	TransQuest
system score	0.8555	0.6642	0.5651	0.7346
std	0.1548	0.1841	0.4232	0.0155
median	0.8875	0.6720	0.6941	0.7368
min	0.0	0.0	-2.4113	0.6548
max	1.0	1.0	1.3308	0.7759

Table 3.3: ModernMT corpus scores distribution

difference in the scores between unseen and seen TUs that were already seen by the system. To test this hypothesis, three recently published texts available on the Globalvoices website in both English and German have been manually selected, extracted and segmented, thus ensuring the quality of the data. Globalvoices is an international, non-profit project, whose community includes writers, translators and activists aiming to translate and report on news from all over the world from the perspective of both media and single citizens[13]. It was selected because of the similarity of its texts to those of the News corpus (see Section 3.3.1) and because one of the corpora of the OPUS collection is composed of texts from the Globalvoices website (Tiedemann, 2012).

[13] `https://globalvoices.org/about/`

	cushLEPOR	BERTScore	COMET	TransQuest
system score	0.8604	0.6619	0.5842	0.7394
std	0.1102	0.1733	0.4003	0.0107
median	0.8789	0.6713	0.7135	0.7402
min	0.0	0.0859	-1.2613	0.7031
max	1.0	1.0	1.1462	0.7676

Table 3.4: Globalvoices corpus scores distribution

	cushLEPOR	BERTScore	COMET	TransQuest
training	0.8875	0.6720	0.6941	0.7368
Globalvoices	0.8789	0.6713	0.7135	0.7402

Table 3.5: Median values for comparison between the training dataset and the Globalvoices dataset.

The corpus includes three texts: "Could the breakdown in public trust explain Hong Kong's sluggish vaccine roll-out?"[14], "Women of colour endure discrimination in Austria's gynecological care"[15] and "Why I might not go back to El Salvador"[16]. All texts have been manually aligned with their German translation and underwent the same preprocessing, translation and scoring steps as the original corpus and includes a total of 128 TUs. Figure 3.2 and 3.1 show the distributions of the 4 metrics divided between the two systems.

A Mann-Whitney U test on all 4 independent variables showed that there was no significant difference (p-value > 0.05, see Table 3.6) between the training dataset compared to the Globalvoices dataset for all metrics except for TransQuest. The median values are provided in Table 3.5.

We thus conclude that there is no significant difference in the scores between our corpus and a corpus containing texts not seen from the MT systems employed. The training corpus will be treated as if all the segments contained have never been seen by the corpus and we will conduct our experiments on the basis of this assumption.

[14] `https://globalvoices.org/2021/05/28/could-the-breakdown-in-public-trust-explain-hong-kongs-sluggish-vaccine-roll-out/` (published June 22, 2021)

[15] `https://globalvoices.org/2021/04/08/women-of-colour-endure-discrimination-in-austrias-gynecological-care/` (published May 5, 2021)

[16] `https://globalvoices.org/2021/01/06/why-i-might-not-go-back-to-el-salvador/` (published October 25, 2021)

	U	p-value
cushLEPOR	749851.0	0.0713
BERTScore	808062.0	0.4736
COMET	764728.0	0.1338
TransQuest	670510.5	0.0004

Table 3.6: Mann-Whitney U Test results

3.4 Architecture

In order to test our research question, we explore two different types of architectures to determine which approach is more suitable for the problem: a single-task and a multi-task model. Being able to leverage the information coming from multiple labels, a multi-task model should be able to offer significantly better predictions when compared to a single-task model, whereas the single-task model would likely have the advantage of requiring much less time for training given the reduced amount of data points.

In the first architecture, we fine-tune and evaluate a transformer model as a single-task model, considering each MT Evaluation and QE metric separately, following the implementation offered by HuggingFace. The second architecture is an extension of the first one, in which we explore the implementation of a multi-task model encompassing all 4 metrics simultaneously, both as a means of transferring knowledge between the metrics and seeking to make the output metric-agnostic. Both architectures employ XLM-RoBERTa as their Encoder (Conneau et al., 2019).

3.4.1 Single-task XLM-RoBERTa

XLM-RoBERTa is a multilingual version of RoBERTa, meaning that it shares with the latter both its pipeline and training objective (Conneau et al., 2019). RoBERTa is a highly performing unsupervised monolingual language model based on BERT, with which it shares the same architecture, but uses a byte-level BPE as a tokenizer (similarly to GPT-2) and a different pretraining scheme. It employs a dynamic masking strategy, in contrast to BERT static masking, where the masking pattern was generated every time a sequence was fed into the model (Conneau et al., 2019).

XLM-RoBERTa extends the language coverage of RoBERTa by means of pre-training on 2.5TB of filtered CommonCrawl data containing 100 languages with a Masked Language Modeling (MLM) objective (See Section 2.2.2.1). Starting from an input sentence, the model randomly masks 15% of the words and passes the masked sentence through the model, which has to predict the masked words. This departs from autoregressive models like GPT which internally mask the future tokens (see Section 2.2.2.2). The model thus obtains a bidirectional representation of the sentence in 100 languages, which can be exploited for the feature extraction with the purpose of fine-tuning the model on different downstream tasks. In our case, having a dataset of source segments with their respective labels, we can implement a regression head on top of the encoder, meaning it would be fed the features produced by XLM-RoBERTa as inputs, in order to fine-tune the model to obtain a given score starting from the source sentences alone.

For our architecture we implement `xlm-roberta-base` in the version made available by HuggingFace (Wolf et al., 2020)[17]. All the experiments are carried out with a learning rate of 2e-5 and using the AdamW optimizer. We explore an effective training batch size $\in [2, 16, 32]$ and epochs $\in [1, 5, 10]$, as suggested for XLM-RoBERTa by a recent study on the performance of multilingual language models by Hu et al. (2020) in their official Github repository[18]. Additionally, we chose to avoid using the predefined loss functions made available by the transformers library, namely Mean Squared Error (MSELoss) and Mean Absolute Error (L1Loss), in favor of the more adaptable HuberLoss provided by the PyTorch library.[19]

This loss combines the advantages of both the Mean Squared Error and the Mean Absolute Error, because it employs a squared term if the absolute element-wise error falls below a predefined delta value (δ) and a delta-scaled Mean Absolute Error otherwise. This characteristic makes the loss less sensitive to outliers than the normal MSELoss, since it treats the errors as squared only if they fall within the delta interval.

[17] `https://huggingface.co/xlm-roberta-base`. Last consulted on 04/02/2022.

[18] `https://github.com/JunjieHu/xtreme-dev/issues/2`. Last consulted on 04/02/2022.

[19] `https://pytorch.org/docs/stable/generated/torch.nn.HuberLoss.html`. Last consulted on 04/02/2022

The HuberLoss for a single prediction is defined as:

$$L_\delta = \begin{cases} \frac{1}{2}(y_i - x_i)^2 & if\ |(y_i - x_i)| < \delta \\ \delta((y_i - x_i) - \frac{1}{2}\delta) & otherwise \end{cases}$$

where x_i is the predicted value and y_i is the corresponding true value.

3.4.2 Multi-Task Learning with a Shared Encoder

In addition to attempting to learn each of our 4 metrics independently, we also experiment with Multi-Task Learning (Caruana, 1997) to link the various label representations together instead of training separate models. This methodology falls within the domain of data augmentation, which aims at improving a model by artificially augmenting the already available data. The idea at the core of multi-task learning is closely linked to the real world, where in reality knowledge is not compartmentalized but linked together. Consequently, attempting to approach a multi-faceted problem as a single task might not be the best approach available and we would instead obtain a better performance by using multiple labels simultaneously. Hence, for closely-related tasks, multi-task learning can leverage the additional data from other tasks to achieve inductive transfer and substantially improve the performance of the model for the single task (Caruana, 1997). This approach has been applied to multiple areas of NLP, ranging from the estimation of the check-worthiness of claims in political debates (Vasileva et al., 2019), to a demographic classifier based on features extracted from tweets (Vijayaraghavan et al., 2017) and fine-tuning of transformer models to improve performance on the GLUE benchmark (Mahabadi et al., 2021).

In our use-case, while in the single-task scenario an independent model is trained for each metric separately, in the multi-task scenario we train multiple models simultaneously, learning the tasks together in a single shared encoder with four different outputs, one for each label (see Figure 3.3 for an abstract representation of the structure). This encoder is mapped across the different tasks, meaning that four separate replicas of the same encoder are created, all sharing the same internal parameters. Once the backpropagation step is performed, all the weights of the four encoders are updated in parallel, thus effectively transferring the information learned from one task to the other.

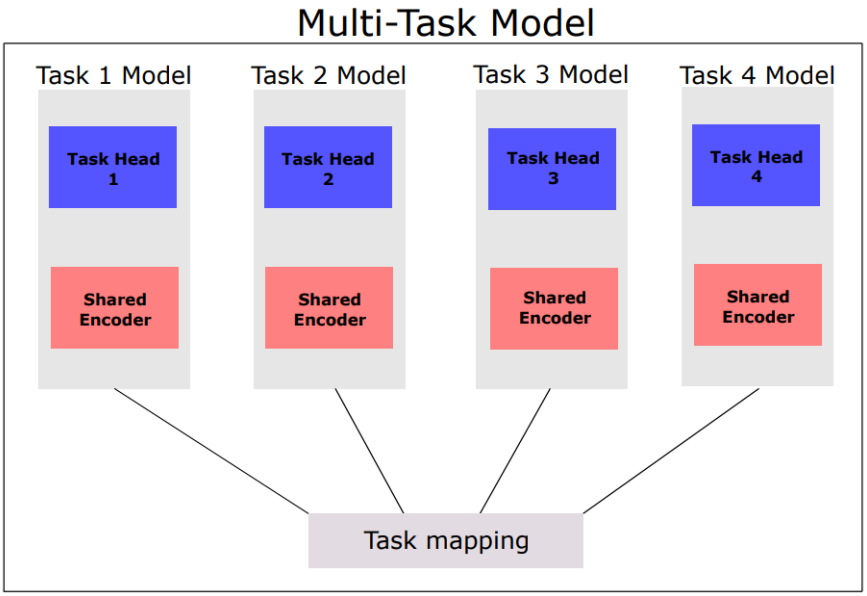

Figure 3.3: Multi-Task model visualization

We test this architecture using the same parameters as the single-label architecture. The major difference is the effective training batch size, which is only set to 2 due to the computational cost of the model and to hardware limitations.

3.4.3 Evaluation

For the evaluation of our model we use the Root Mean Square Error (RMSE), in the implementation made available by `sklearn`[20]. It is a more easily interpretable version of the Mean Squared Error, representing the standard deviation of the prediction errors. It expresses the average distance of the predictions from the golden standard, thus it informs the reader on how similar the output data is compared to the original predictions. Since it is scale-dependent, it cannot be interpreted as an absolute measure across distributions, meaning it is only informative with respect to the original distribution and we cannot compare it across multiple tasks. In our case, this

[20] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html`. Last consulted on 07/02/2022.

means that each evaluation carried out using RMSE will not be comparable with the other metrics, since the underlying distributions of the gold standard are different. RMSE is computed as:

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2}$$

where x_i is the predicted value of the -ith sample, and y_i is the corresponding true value. The Root Mean Squared Error is then estimated over n samples.

Chapter 4

Experiments

4.1 Introduction

This Chapter focuses on the experiments performed on the basis of the experimental settings established in Chapter 3.

Section 4.2 provides an overview of the results, including all combinations of the settings explored as well as the details regarding model training.

Section 4.3 offers an analysis of the results, answering the research questions posed in the previous Chapter (see Section 3.2). It also tries to offer a perspective on the limitations of this dissertation and its practical applications, underlining the key aspects to address in future research.

4.2 Results

This Section offers a report on the performance obtained by both the single-task and multi-task model. The reported score is the Root Mean Squared Error (RMSE) (see Section 3.4.3), computed over the predictions obtained by the models on the 1584 segments of the test set (see Section 3.3.1). Since the distributions of the labels fall within different ranges, depending on the characteristics of each metric, the RMSE value is not comparable across tasks. This characteristic makes the metric only informative with respect to the original distribution. Thus, in our use-case, all model predictions and gold labels have been reshaped to the $[0, 1]$ range, in order to obtain a value

	cushLEPOR	BERTScore	COMET	TransQuest
single@1	**0.1517**	**0.1617**	0.1475	0.3155
multi@1	0.1553	0.1693	**0.1468**	**0.1205**
single@5	0.1548	0.1651	**0.1373**	0.4327
multi@5	**0.1538**	**0.1609**	0.1459	**0.1052**
single@10	0.1643	0.1690	0.1490	0.3616
multi@10	**0.1547**	**0.1593**	**0.1462**	**0.1102**

Table 4.1: Results using a training batch size of 2 and evaluated after [1; 5; 10] epochs. The score is reported as normalized RMSE value. Best result on epochs is in bold and best result on metric is in red.

which is not only comparable but also easily interpretable across different tasks. A lower score corresponds to a better performance, since it indicates that the predictions were closer to the gold standard. Table 4.1 shows the results for both the single-task and multi-task XLM-R model, using a learning rate of 2e-5 and batch size of 2. The training was carried out using a NVIDIA Quadro P4000 8GB GPU. It lasted 6 hours for each single-task model and 32 hours for the multi-task model.

At 1 epoch the single-task and multi-task models obtain very similar scores when predicting cushLEPOR, BERTScore and COMET. Although the single-task model has a slightly better performance on the first two metrics, it is outperformed by the multi-task model on TransQuest, which is an unexpected result.

Upon closer inspection of the actual predictions, it is clear that in reality the single-task model is converging to the mean, effectively only predicting the mean value (0.7413) for all instances and not generating a real prediction for the segment.

At 5 epochs the results are fundamentally unchanged, although all scores for the multi-task model show a slight improvement and surpass the single-task model for both cushLEPOR and BERTScore, since the latter seems to begin degrading a little on both metrics. For COMET the situation also changes, with the single-task model instead showing signs of improvement. TransQuest instead shows the overall worst degradation, further distancing itself from the multi-task model, which instead obtains the best RMSE value across all tests.

	cushLEPOR	BERTScore	COMET	TransQuest
single@1	0.1317	**0.1728**	0.1086	**0.1943**
single@5	**0.1250**	0.1785	0.1082	0.3334
single@10	0.1444	0.1842	**0.1056**	0.2980

Table 4.2: Results using a training batch size of 16 at different epochs, only using single-task models. The score is reported as normalized RMSE value and the best performances are highlighted in bold.

	cushLEPOR	BERTScore	COMET	TransQuest
single@1	0.1325	0.1728	**0.1076**	**0.1788**
single@5	**0.1089**	**0.1704**	0.1083	0.2612
single@10	0.1498	0.1833	0.1105	0.3132

Table 4.3: Results using a training batch size of 32 at different epochs, only using single-task models. The score is reported as normalized RMSE value and the best performances are highlighted in bold.

At 10 epochs all models start degrading, but the multi-task version seems to be more resistant in this regard. It outperforms the single-task model on all 4 tasks, albeit only by a slight margin, remaining fundamentally stable in its performance. Although the TransQuest single-task model seems to show signs of improvement, in reality the model at 10 epochs behaves similarly to the model at 1 epoch, since it, too, only predicts the mean value regardless of the segment.

Tables 4.2 and 4.3 show the results for the single-task XLM-R models using the same learning rate as before but exploring a batch size of 16 and 32, respectively. The training was carried out using the same GPU as before, but it lasted 4 hours for the models using a batch size of 16 and 2:30 hours for the models using a batch size of 32. As mentioned in Section 3.4, there is no multi-task model for this batch size due to the computational cost of loading multiple encoders at the same time on a single 8GB GPU.

All models trained on a batch size of 16 exhibit a trend whereby they obtain a better score when compared to the models trained on the smaller batch size. In particular, for COMET the score always hovers around 0.108 while in previous instances it never went below 0.137. For this group the

TransQuest model also shows improvement, managing to obtain a score 0.194 already in the first epoch, which is significantly lower than the previous model in Table 4.1, although it degrades at later epochs. Both BERTScore and cushLEPOR remain relatively stable across epochs, with the former obtaining a slightly worse performance and the latter significantly improving on the model with a smaller batch by almost 0.03 points at 5 epochs. This time the TransQuest model shows no tendency to regress to the mean for all epochs.

Models trained on a batch size of 32 exhibit a similar trend to those trained on a batch size of 16. They obtain a better score when compared to the models trained on the smaller batch size, but this improvement is only noticeable for the cushLEPOR and TransQuest model. The cushLEPOR model in particular obtains almost 0.05 points less than the model with a batch size of 2, thus obtaining the lowest score observed so far, while the TransQuest model further improves at only 1 epoch with a score of 0.179 before degrading. Once again, the score for COMET always hovers around 0.108 and the BERTScore model performs in line with the one shown in Table 4.2.

4.3 Discussion

With respect to the research questions posed in Section 3.2, the results reported in the previous Section are indeed promising. As a recap, the research questions are the following:

> ***Research Question 1:*** *Is it possible to accurately predict the MT Evaluation or Quality Estimation score from the source text alone?*
>
> ***Research Question 2:*** *If the first research question is true, is it better to construe the problem as a single-task or as a multi-task problem?*

Given that, on average, the reported RMSE value ranges between 0.1 and 0.2, all models do seem to be able to predict a metric of translation fairly accurately starting from the source text alone, which corroborates the first

research question. The only exception concerns the TransQuest single-task model using a batch size of 2, which regresses to the mean at epochs 1 and 10, while at 5 the performance is definitely subpar when compared with both the multi-task model and the single-task models with higher batch sizes (see Tables 4.1, 4.2 and 4.3). This could be caused by the particularly small range for the score distribution of the TransQuest metric, which is bound between $[0.65, 0.78]$ (see Table 3.3). Additionally, since almost all single-task systems seem to start degrading at higher epochs, overtraining might also be an issue, leading the model to fail to correctly predict the metric. Overall, for this use-case the best epoch configuration seems to be at 5, since this is the interval when the models either obtain the best score – such as multi@5 on TransQuest (see Table 4.1) or single@5 on cushLEPOR (see Table 4.3) — or do not distance themselves significantly from the best score — such as single@5 against single@1 on COMET (see Table 4.3) and against single@10, again on COMET (see Table 4.2). The single-task models on TransQuest are once again the exception, since all models start degrading after the first epoch. Regarding batch size, as expected, there is an improvement in the performance of all models when it is increased from 2 to 16, except for the predictions for BERTScore, which observe a slight degradation. The difference between a batch size of 16 and 32 is less pronounced, but it does allow the model to reach the best performance on cushLEPOR with single@5 (see Table 4.3). Thus, it is possible to conclude that for the single-task model the best performing batch size is 32, also thanks to its reduced training time, though it is indeed more costly in terms of memory requirements. Considering all of the above, we conclude that the first research question is corroborated by the results obtained by both the single-task and multi-task models, meaning that it is possible to accurately predict evaluation scores from the source text.

Regarding the second research question, the answer is not as straightforward. There indeed is an improvement in the performance for the multi-task model, especially on TransQuest, which shows that knowledge transfer occurs when training on multiple metrics. Nevertheless, the results for all other metrics are overall constant, showing no noticeable sign of improvement past the 5 epoch margin (see Table 4.1). Based on the findings from the previous research question and given its stability, it would be possible to assume that with a bigger batch size the model benefits from seeing multiple segments at once before the backpropagation step. Researching higher batch sizes would thus be the natural following step to the current study. One major concern,

though, is the training time required to achieve these results for a multi-task model. Many recent studies on sustainability in NLP have pointed out a major issue with the trends towards the usage of large transformer models. Since their carbon footprint is increasingly impactful, the research community is highlighting the need to prefer lighter models with smaller training times when solving an NLP task (Strubell et al., 2019; Anthony et al., 2020; Bannour et al., 2021). As pointed out in the beginning of Section 4.2, the multi-task model used for this study took around 32 hours to train, much longer than a single-task model, which took a fifth of the time each, around 6 hours, which was further decreased to 2:30 hours when scaling to higher batch sizes. Although the training time for the multi-task model can be decreased by using a bigger batch size, so far the current results do not justify the usage of such an expensive multi-task model for this task, since the performance of the single-task model does not distance itself significantly from the multi-task version. Hence, with respect to the second research question we conclude that, while a multi-task model shows promising results for this task, it is better to construe the problem as single-task, because this method offers the best trade-off in terms of training time and performance.

Summing up, we have observed that the low error margins shown in these experiments point towards the possibility to estimate the quality of MT based on the source text alone. This indicates that the initial hypothesis that motivated this study is corroborated and it is possible to assert that there indeed is a direct connection between the source text and the performance of an MT system, as measured by well-established evaluation metrics (see Section 3.2).

Nevertheless, it is imperative to also underline two key limitations of this study. First and foremost, the corpus which was used has likely been already seen by the MT systems. This is a significant weakness of the study, although a set of preliminary experiments has shown no significant difference between unseen and seen texts (see Section 3.3.1). In order to overcome this potential limitation, one would need to either build a dedicated parallel corpus on which to apply the proposed methodology, or train an MT engine from scratch, both of which are beyond the scope of this work. The second limitation is the absence of a pipeline for terminology recognition in the source text. Given the importance of terminology in the field of translation, such a component would indeed be invaluable to correctly assess machine translation suitability and improve model performance, while also ensuring

the correctness of the predictions (Cabré, 2010; Scansani et al., 2019, 2017).

In conclusion, although the models described in this work are certainly not production-ready, they provide a good starting point for multiple real-life applications. Having a tool to predict MT quality from the source text is of increasing interest for the industry. One such example is a recent proposal for the development of a pipeline to predict the impact of source text on machine translation at the MTSummit2021, which showed very promising results using solely textual features (Yanishevsky, 2021) (see Section 2.4). The models in our study could be used in a similar fashion to check for linguistic quality risks, leading to significant savings in terms of time and costs when performing linguistic quality assessment. By integrating them in the APE-QUEST pipeline it would be possible to skip the initial requirement to request a machine translated version of the text, only performing it when needed, thus avoiding unnecessary calls to the MT system. Another application could involve identifying the highest quality segments to customise an MT engine, or conversely to find the most challenging ones for the evaluation step. Given the plethora of possible applications, further research on these methodologies is warranted, since the methods proposed show good promise and significant improvements could certainly prove beneficial to the translation world, both from the perspective of research and of the industry.

Chapter 5

Conclusions

This dissertation attempted to answer two research questions, i.e., "Is it possible to accurately predict the MT Evaluation or Quality Estimation score from the source text alone?" and "If the first research question is true, is it better to construe the problem as a single-task or as a multi-task problem?". Both are based on the hypothesis that 'The better the machine translated version of a text, in terms of a well-established evaluation metric, the more suitable is the source text to machine translation'. The questions were motivated by the increasing need to automatically assess the quality of machine translation in a way that is both dynamic and scalable, without the limitation of providing very expensive reference translations. While there exists a field entirely dedicated to reference-less metrics, namely Quality Estimation, this work tried to explore innovative techniques that would focus entirely on the source text. Such an approach offers an alternative that could, in the future, also account for the pivotal issue of terminology. To the best of my knowledge, this work is the first attempt of predicting an evaluation score for machine translation suitability starting from the source text alone, without requiring to have a machine-translated version of the segment.

The corpus at the basis of this dissertation was created by collecting and cleaning out-of-the-box bilingual `en>de` corpora from the OPUS collection. The starting corpora were translated using two machine translation engines (ModernMT and Microsoft Translator) and merged together in order to obtain an evaluation score for each segment, using both traditional evaluation and quality estimation metrics. After their calculation, each source segment was paired with its respective scores, thereby creating a corpus of source segments with four different labels. This corpus was subsequently used to

fine-tune a transformer model (XLM-RoBERTa) both in a single-task and a multi-task setting. The scripts used for the experiments are available on GitHub[1]. By obtaining an RMSE score as low as 0.10, the current results are indeed promising and answered positively to the first research question, indicating that the multi-task model might be very well-suited for this task, albeit there remain concerns regarding the computational cost and sustainability issues of the multi-task model. Nevertheless, these indicated that it is indeed possible to obtain accurate machine translation evaluations starting from the source text alone, paving the way for further research in this regard.

Future research could improve many aspects touched by this dissertation. Since XLM-RoBERTa is a multilingual model, an initial focus could be posed on extending the experiments to other language pairs, surveying significant differences among different language combinations and directions. Additionally, a pipeline for terminology recognition in the source text would certainly offer valuable information for the final prediction, while a connection with research on source text translatability for humans could perhaps provide further insights on the overall problem. From a technical point of view, two main aspects could undoubtedly be improved in order to surpass the current limitations: the training corpus and the training methodology. As stated in Chapter 4, in order to ensure the validity of these conclusions, one should make sure that the training corpus has not been seen by the MT systems. Further research could thus improve on this aspect by either creating an ad-hoc parallel corpus on which to compute the initial scores or by computing the translation using a different MT engine, where one could ascertain the texts on which it was trained. For what concerns the training methodology, the experiments could be extended by using a bigger batch size for the multi-task model — which following the current results should lead to both faster training and performance boost — exploring different epoch configurations or employing different transformer models.

[1] `https://github.com/TinfFoil/MTsweet`

Bibliography

Mohamed Alloghani, Dhiya Al-Jumeily, Jamila Mustafina, Abir Hussain, and Ahmed J Aljaaf. A systematic review on supervised and unsupervised machine learning algorithms for data science. *Supervised and unsupervised learning for data science*, pages 3–21, 2020.

Lasse F Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*, 2020.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Nesrine Bannour, Sahar Ghannay, Aurélie Névéol, and Anne-Laure Ligozat. Evaluating the carbon footprint of nlp methods: a survey and analysis of existing tools. In *EMNLP, Workshop SustaiNLP*, 2021.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. *arXiv preprint arXiv:1608.04631*, 2016.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based mt quality: An in-depth analysis on english–german and english–french. *Computer Speech & Language*, 49:52–70, 2018.

Nicola Bertoldi, Davide Caroselli, and Marcello Federico. The modernmt project. *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, page 345, 2018.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn,

Varvara Logacheva, Christof Monz, et al. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, 2016.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Maria Teresa Cabré. Terminology and translation. *Handbook of translation studies*, 1:356–365, 2010.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Sheila Castilho, Stephen Doherty, Federico Gaspari, and Joss Moorkens. Approaches to human and machine translation quality assessment. In *Translation Quality Assessment*, pages 9–38. Springer, 2018.

Noam Chomsky. On certain formal properties of grammars. *Information and control*, 2(2):137–167, 1959.

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. *arXiv preprint arXiv:1911.02116*, 2019.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Heidi Depraetere, Joachim Van den Bogaert, Sara Szoc, and Tom Vanallemeersch. Ape-quest: a quality gate for routing mt. In *Proceedings of the European Association for Machine Translation 2020*, pages 473–474, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondrej Bojar. Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Sixth Conference on Machine Translation, Association for Computational Linguistics, Online*, pages 10–11, 2021.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://aclanthology.org/W13-2305`.

Aaron L. F. Han, Derek F. Wong, and Lidia S. Chao. LEPOR: A robust evaluation metric for machine translation with augmented factors. In *Proceedings of COLING 2012: Posters*, pages 441–450, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL `https://aclanthology.org/C12-2044`.

Lifeng Han, Irina Sorokina, Gleb Erofeev, and Serge Gladkoff. cushLEPOR: customising hLEPOR metric using optuna for higher agreement with human judgments or pre-trained language model LaBSE. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1014–1023, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.109`.

Hannes Hapke, Cole Howard, and Hobson Lane. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python.* Simon and Schuster, 2019.

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

Charles F Hockett and Charles D Hockett. The origin of speech. *Scientific American*, 203(3):88–97, 1960.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR, 2020.

W John Hutchins and Harold L Somers. An introduction to machine translation, 1992.

Julia Ive, Lucia Specia, Sara Szoc, Tom Vanallemeersch, Joachim Van den Bogaert, Eduardo Farah, Christine Maroti, Artur Ventura, and Maxim Khalilov. A post-editing dataset in the legal domain: Do we underestimate neural machine translation quality? In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3692–3697, 2020.

Thorsten Joachims. 11 making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, page 169, 1999.

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.

Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

Philipp Koehn. *Neural machine translation*. Cambridge University Press, 2020.

Philipp Koehn, Franz J Och, and Daniel Marcu. Statistical phrase-based translation. Technical report, University of Southern California Marina Del Rey Information Sciences Inst, 2003.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, 2020.

Sotiris B Kotsiantis, I Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463, 12 2014. doi: 10.5565/rev/tradumatica.77.

Matouš Macháček and Ondřej Bojar. Results of the WMT13 metrics shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://aclanthology.org/W13-2202`.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*, 2021.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *arXiv preprint arXiv:2106.15195*, 2021.

Nitika Mathur, Tim Baldwin, and Trevor Cohn. Tangled up in BLEU: reevaluating the evaluation of automatic machine translation evaluation metrics. *CoRR*, abs/2006.06264, 2020. URL `https://arxiv.org/abs/2006.06264`.

MD Okpor. Machine translation approaches: issues and challenges. *International Journal of Computer Science Issues (IJCSI)*, 11(5):159, 2014.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040`.

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. doi: 10.18653/v1/W18-6319. URL `https://aclanthology.org/W18-6319`.

Chiew Kin Quah. *Translation and technology*. Springer, 2006.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020a.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, 2020b.

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. doi: 10.18653/v1/2020.emnlp-main.213. URL `https://aclanthology.org/2020.emnlp-main.213`.

Ehud Reiter. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401, 2018.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL `https://aclanthology.org/2020.tacl-1.54`.

Randy Scansani, Silvia Bernardini, Adriano Ferraresi, Federico Gaspari, and Marcello Soffritti. Enhancing machine translation of academic course catalogues with terminological resources. In *Proceedings of the Workshop Human-Informed Translation and Interpreting Technology*, pages 1–10, Varna, Bulgaria, September 2017. Association for Computational Linguistics, Shoumen, Bulgaria. doi: 10.26615/978-954-452-042-7_001. URL `https://doi.org/10.26615/978-954-452-042-7_001`.

Randy Scansani, Luisa Bentivogli, Silvia Bernardini, and Adriano Ferraresi. MAGMATic: A multi-domain academic gold standard with manual annotation of terminology for machine translation evaluation. In *Proceedings of*

Machine Translation Summit XVII: Research Track, pages 78–86, Dublin, Ireland, August 2019. European Association for Machine Translation. URL https://aclanthology.org/W19-6608.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas. URL https://aclanthology.org/2006.amta-papers.25.

Lucia Specia. Fundamental and new approaches to statistical machine translation. *Wolverhampton: Unviersity of Wolverhampton*, 2010.

Lucia Specia and Kashif Shah. Machine translation quality estimation: Applications and future perspectives. In *Translation Quality Assessment*, pages 201–235. Springer, 2018.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.79.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wmt-1.71.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, pages 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

Shuo Sun, Francisco Guzmán, and Lucia Specia. Are we estimating or guesstimating translation quality? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6262–6267, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21, 2020.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

Tom Vanallemeersch and Sara Szoc. Ape-quest, or how to be picky about machine translation? *Translating and the Computer 42*, page 84, 2020.

Bram Vanroy, Orphée De Clercq, and Lieve Macken. Correlating process and product data to get an insight into translation difficulty. *Perspectives*, 27 (6):924–941, 2019.

Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. *arXiv preprint arXiv:1908.07912*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Prashanth Vijayaraghavan, Soroush Vosoughi, and Deb Roy. Twitter demographic classification using deep multi-modal multi-task learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 478–483, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2076. URL https://aclanthology.org/P17-2076.

Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. Progress in machine translation. *Engineering*, 2021. ISSN 2095-8099. doi: https://doi.org/10.1016/j.eng.2021.03.023. URL https://www.sciencedirect.com/science/article/pii/S2095809921002745.

Philip Williams, Rico Sennrich, Matt Post, and Philipp Koehn. Syntax-based statistical machine translation. *Synthesis Lectures on Human Language Technologies*, 9(4):1–208, 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Alex Yanishevsky. Bad to the bone: Predicting the impact of source on MT. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 175–199, Virtual, August 2021. Association for Machine Translation in the Americas. URL https://aclanthology.org/2021.mtsummit-up.14.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SkeHuCVFDr`.