FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Iryna Tryhubyshyn

Mutual Relation of Machine Translation and Quality Estimation

Institute of Formal and Applied Linguistics

Supervisor of the master thesis:	Mgr. Aleš Tamchyna, Ph.D.
Consultant:	doc. RNDr. Ondřej Bojar, Ph.D.
Study programme:	Computer Science
Study branch:	Artificial Intelligence

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....
Author's signature

I would like to express my gratitude to my thesis supervisor, Mgr. Aleš Tamchyna, Ph.D. and to my advisor, doc. RNDr. Ondřej Bojar, Ph.D. for their patience, time and their guidance. I also thank my partner Luka for his support. Without him, this thesis would not have been possible.

Title: Mutual Relation of Machine Translation and Quality Estimation

Author: Iryna Tryhubyshyn

Department: Institute of Formal and Applied Linguistics

Supervisor: Mgr. Aleš Tamchyna, Ph.D., Memsource

Consultant: doc. RNDr. Ondřej Bojar, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Machine Translation Quality Estimation predicts quality scores for translations produced by Machine Translation systems based on source and output segments. Quality Estimation systems are usually trained in a supervised manner using training data that contains translation produced by one or more (other) Machine Translation systems. Therefore, the choice of training data for Machine Translation has an impact on how well the Quality Estimation system works.

This thesis studies the relationship between Machine Translation systems and sentence-level Quality Estimation systems. Using our definitions of Machine Translation system power and Quality Estimation system power, we conducted experiments that involve training Machine Translation and Quality Estimation systems of varying power. We presented Quality Estimation systems evaluation results on test sets of different domains and translated by Machine Translation systems of different power. We find that (i) Quality Estimation systems trained on translations of lower quality outperform Quality Estimation systems trained on translations of higher quality; (ii) evaluating high-quality Machine Translation systems is challenging for Quality Estimation systems of all powers; (iii) high-power Quality Estimation systems work better for out-of-domain distribution than low-power Quality Estimation systems.

Keywords: machine translation quality estimation machine learning deep learning

Contents

Introduction

Machine Translation Quality Estimation (MT QE) is the task of predicting how good the machine translation is without access to a reference translation. Quality Estimation works on top of Machine Translation. Supplied with a translation generated by the Machine Translation system and the source segment, the Quality Estimation model assigns the quality score to the segment. Quality Estimation can be applied on different levels of granularity when segments could be separate words and phrases or whole sentences and documents. Segment quality is the score assigned to the segment by some evaluation procedure. Depending on which evaluation procedure QE aims to estimate, there are different variations of the models. For example, models can estimate post-editing effort or direct assessment scores.

Quality scores predicted by the Quality Estimation system might be the end product displayed to the user. Machine Translation systems do not always work smoothly. For some input sentences, translators generate an excellent translation; in other cases, the translation contains mistakes or is incorrect. Quality scores can tell the user if the particular translation can be trusted or not.

Quality Estimation system can also help to prepare datasets for Machine Translation. They can be used to filter out low-quality sentences in datasets. The Quality Estimation system can preselect the data for manual annotation. Using Quality Estimation systems, the researchers can choose the data with a specific quality distribution. For example, when selecting the dataset for post-edition, they can exclude high-quality translations that do not need post-editing.

Quality estimation is usually done in a supervised manner. The train datasets consist of the source sentences, translations generated by the Machine Translation system and target quality scores. Therefore, each Quality Estimation system is bound to the Machine Translation system that produces translation for training. It is not clear how exactly the system performs the estimation. It is possible that the system needs to know how the translation is performed. In this case, the estimation involves a process that resembles translation. QE learns how to perform this process from the translation supplied in training. If simple QE with small capacity is paired with large, high-performing MT, QE should struggle to extract the translation knowledge because it would not fit into the QE capacity. Moreover, if QE is trained on the low-quality data, it will not have enough knowledge to grade the higher-quality data. On the other side, there might be features that are good indicators of translation quality and do not require translation knowledge. Then small QE that knows these heuristics will perform well even if it was trained on the data from high-performing MT.

This thesis studies how changes in translation data and QE model capacity affect QE model performance. We train the MT system on different amounts of data to get several systems of varying power. We use datasets generated by those systems to train QE systems. The goal of this thesis is to train QE models on the data from various MT changing the data capacity and study how the model's performance depends on this data. We evaluate QE systems on multiple datasets to check their behavior under the domain shift.

Structure

Chapter 1 is dedicated to the Machine Translation. We explain how the modern Machine Translation system works. We describe approaches to Machine Translation evaluation and how the power of a Machine Translation system is defined.

Quality estimation is covered in chapter 2. We describe the task of Quality Estimation and the existing solution to the task. We explain how QE systems are connected with MT. We describe related finding on the relation between MT capacity and QE capacity strength.

Datasets and tools are covered in chapter 3. We describe the dataset used in this work and how we preprocessed them to create QE dataset. Then we talk about the MT systems used in experiments and how we trained them. We give details about QE systems and their training.

Chapter 4 presents the experiments. We explain why we conducted the experiments. We present the evaluation results on the test set. We talk about what conclusion should be made from these results.

1. Machine translation

This chapter focuses on machine translation task. We start by describing the task's history and Statistical Machine Translation and Neural Machine Translation approaches to the task. We discuss how the modern MT systems work using Transformer architecture as an example. Then we talk about Machine Translation evaluation. We explain what we mean under the power of Machine Translation systems and how data and model capacity contribute to the power of the MT system.

Machine Translation (MT) is the task of automatically translating text from one language to another. Machine Translation history is similar to the history of artificial intelligence as a field: it starts with rule-based systems, then shifts to classical machine learning, and then to deep learning. We can divide the history of machine translation into three periods:

- Up to the 1990s. Period of rule-based Machine Translation. The general idea is to translate sequences in a deterministic way using dictionaries and rules created using linguistic information about language grammar.

- 1990s - early 2010s. The period when statistical Machine Translation dominated the field.

- From the mid-2010s. The rise of neural Machine Translation.

1.1 Statistical Machine Translation

The idea behind Statistical Machine Translation (SMT) is that computers can learn how to translate automatically from examples of translations instead of being programmed by people. That introduces us to the concepts of training and inference. During training, models are supplied with data and retrieve information on how to translate. The inference is a process of translation when already trained models are used to produce a translation. Generally, training a good model requires many data. The shift to SMT was possible because of the availability of large parallel datasets.

These methods give us more flexibility because, in principle, we can use the same model with different languages simply by changing training data. They give us two basic directions on how we can improve Machine Translation. We can either work on the model to make it better capture the knowledge from data or improve the data by increasing data size or data quality.

The term Statistical Machine Translation refers to a variety of methods. Its basic idea is that we want to get the model that finds the most probable translation given the source sentence. One of the popular methods is phrase-based Machine Translation (Koehn et al. [2003]). In its basic form, it consists of a phrase-based translation model and language model used to translate phrases and an alignment mechanism to reconstruct the sentence in other languages taking into account differences in word order.

While SMT is rarely used nowadays, many methods developed at that time are still used today, for instance, metrics for MT evaluation and approaches for manual evaluation. Moreover, the whole Quality Estimation task that is the main focus of this thesis emerged in the SMT era.

1.2 Neural Machine Translation

Modern MT systems are predominantly neural network-based. They arose in the mid-2010s when deep learning was gaining popularity. Important milestones in NMT include development of Seq2seq architecture (Sutskever et al. [2014]) and Transformer architecture (Vaswani et al. [2017]). Solution based on Seq2seq architecture was the first neural network-based solution that Google put in production (Wu et al. [2016]). The Transformer beat Seq2seq in performance, became a new state of the art, and nowadays is the most popular architecture for NMT tasks.

Seq2seq and Transformer implement the encoder-decoder architecture. These systems typically operate on the sentence level. Each inference run model produces a prediction for the next token in the output sentence. The output is produced token by token. The model consists of two parts: encoder and decoder. Encoder transforms source sentence into an intermediate representation. The decoder consumes that intermediate representation and already produced part of the translation, and predicts the next token.

The model works with tokenized sentences. The vocabulary should be known beforehand. Tokenization is handled outside of the model training. Tokenization can be performed on word, subword, or character level. Word level tokenization brings the issue of out-of-vocabulary words. Subword tokenization splits rare words into subwords which helps with the out-of-vocabulary issue. Subword level tokenization is more efficient than character level. Sentences tokenized into characters create much longer sequences than subword or word tokenization. That makes training and inference more computation heavy and creates performance problems.

Byte-pair encoding (Sennrich et al. [2016]) is a widely used tokenization method that works on the subword level. The vocabulary is formed by merging operations. At the start, the vocabulary consists only of characters that appear in the dataset. In each merging operation, the most frequent sequence of 2 tokens is merged and added to the vocabulary until the necessary size is reached. The size of the vocabulary is a training hyperparameter. The smaller the vocabulary is, the closer result is to the character-level tokenization. Big vocabulary size enlarges the embedding and the output layers, which increases memory consumption and training time.

Encoder-decoder model produces a probability distribution over the next token in the sequence. The exact translation is generated from this distribution by a decoding algorithm. The typical approach is to yield a sentence with maximum likelihood, which generally is an NP-complete task. The common approximate algorithm used for this task is the beam search algorithm. By this algorithm, the model works with top n predictions simultaneously, where n is a hyperparameter of the algorithm. In each step, the model generates predictions for all n options and selects a new top n from predictions.

1.2.1 Transformers

As an encoder-decoder architecture, the Transformer takes the whole source sentence and output prefix, and produces a probability distribution over the next token in the sequence. The Transformer can be seen as a collection of different blocks. Output shape of all model's blocks is the same which allows to stack blocks one on another and add residual connections. Aside from the input and output layer, there are two kinds of blocks: encoder and decoder blocks. The vanilla Transformer consists of 6 encoder blocks and six decoder blocks. Figure 1.1 shows the transformer scheme that includes a scheme for each block type and attention mechanism scheme.

The input layer, also called embedding, maps input tokens to the vectors. The embedding acts like a lookup table, where keys are tokens from vocabulary, and values are numeric vectors that serve as the numerical representation of their keys. The embedding is applied to the source sentence before the first encoder block and to the translated sentence before the first decoder block. The input to other blocks is a two-dimensional matrix. The first dimension is the sequence size, and the second dimension is the embedding size.

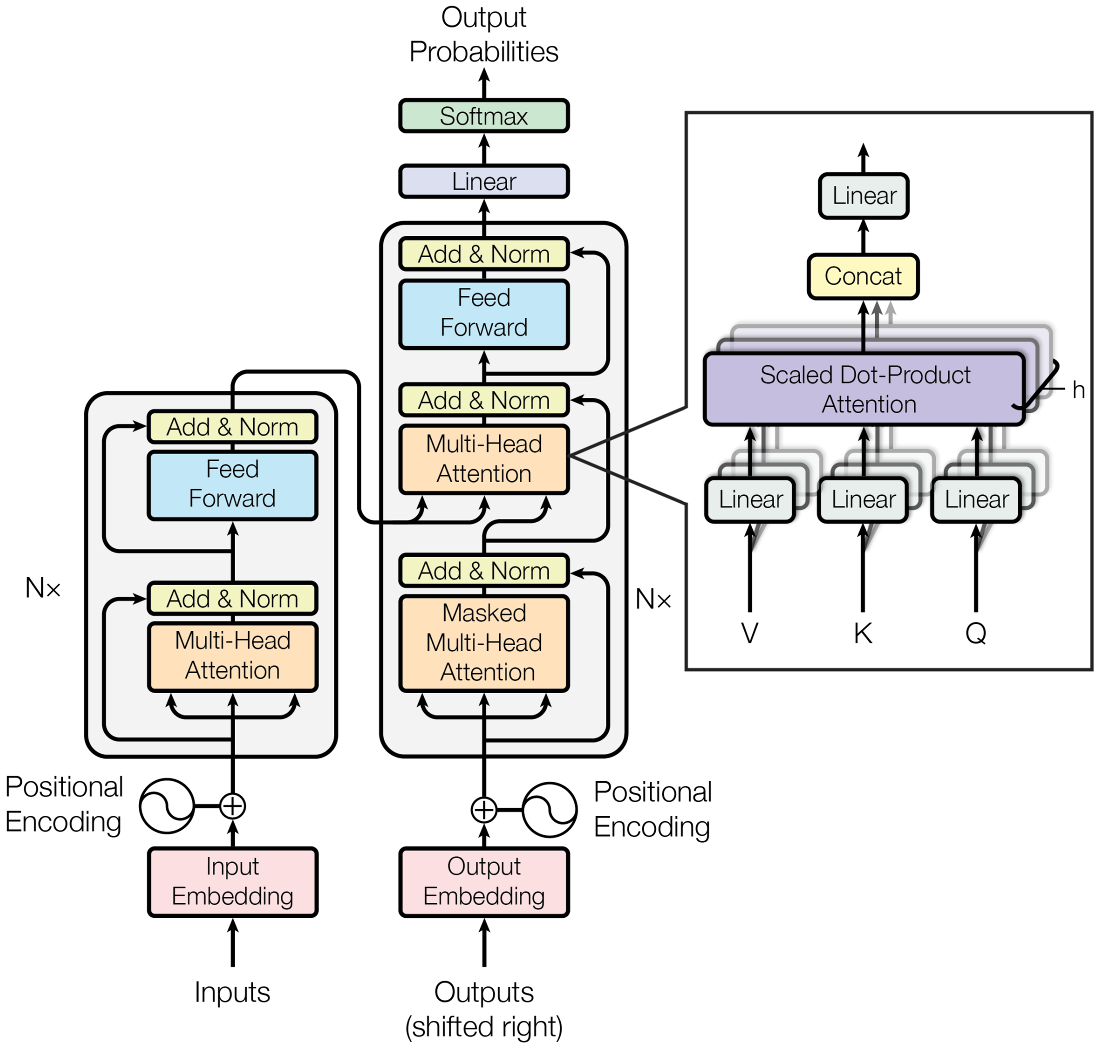

Figure 1.1: Transformer architecture scheme (Vaswani et al. [2017])

The Transformer produces a probability distribution over the vocabulary. The output of the last decoder layer is fed into a fully-connected layer that produces the output of vocabulary size. Then the softmax is applied to obtain probabilities. Embedding and output layers are trainable. The number of their parameters is tied to vocabulary size, which is a training hyperparameter. Since the number of their parameters depends on the training settings, embedding and output layers are ignored when displaying the number of trainable parameters in a Transformer.

The encoder block consists of two sub-layers. The first sub-layer is the self-attention layer. The attention step is the only moment where computation takes into account other tokens in the sequence. The rest of the steps treat each token separately from the rest. The Transformer uses multi-head attention. The second sub-layer is a fully-connected feed-forward layer. After each layer, there is a residual connection and layer normalization.

The decoder block has an additional step before the feed-forward layer. It is a layer that computes attention over the output of the last encoder layer. Self-attention allows the decoder to gather information from the already produced translation sequence, while encoder-decoder attention brings information from the tokens in the source sentence. In self-attention, the decoder can only see tokens left to the current token, which prevents the decoder from looking at the tokens that are not produced yet.

The attention mechanism does not take into account the position of the tokens. This information is supplied to the network by positional encoding. The vector that encodes the sentence's position is summed with the input embedding. It can be computed by some function of the position number or learned in the same way as token embedding.

1.2.2 Attention mechanism

Attention brings context into token representation. Without attention, the final representation of each token would be calculated in isolation from other tokens in the sentence. However, when creating a contextual representation of a token, words in a sentence are not equally important.

Attention computes word representation as a weighted average of all words in the sentence. Weights are computed dynamically. Weight between 2 words represents how related the words are. As a result, related words impact word representation more than unrelated ones.

The attention mechanism consists of 3 elements: Queries, Keys, and Values. They are computed from input vectors x_q, x_k, and x_v using matrices W^q, W^k, W^v:

$$Q = x_q W^q \quad K = x_k W^k \quad V = x_v W^v$$

Weights are computed with a similarity function between Queries and Keys. Values are token representations used in weighted average. In self-attention Queries, Keys and Values are computed from the input vector. In encoder-decoder attention, Queries are computed from the input vector, Keys and Values are computed from the encoder output. The resulting attention is computed as:

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

The idea of multi-head attention is to compute attention multiple times using sets of matrices W^q, W^k, W^v. Heads learn a different set of features independently from each other. The resulting attentions are concatenated and projected using matrix W^v. The formula can be written as:

$$MultiHead(Q, V, K) = Concat(Att_1(Q, V, K), ..., Att_k(Q, V, K))W^O$$

All inputs should be vectors of the same size. If d_x - size of the vector and k - numbers of head then each W^q, W^k, W^v have size $(d_x \times d_x/k)$ and W^o has size $(d_x \times d_x)$. Trainable parameters in attention are matrices W^q, W^k, W^v and W^v.

1.3 Machine Translation evaluation

This section is dedicated to the evaluation of Machine Translation. Before discussing Machine Translation power, we should explain how Machine Translation systems are evaluated. Moreover, quality estimation is the task of creating a system that predicts human evaluation scores. To understand the QE task, we should know what qualities those scores represent.

1.3.1 Manual evaluation

Manual evaluation is the evaluation carried out by humans. Evaluating the translation is not a straightforward task. Machine Translation systems aim to produce a translation that sounds fluent to humans and holds the same meaning as the source text. Whether the translation has those qualities can be decided only by a human. There is no other objective way how to evaluate a translation. The ground truth scores for translation are gathered in manual evaluation.

Annotators should generally evaluate translation only into their native language. If annotators are proficient in the source language, they evaluate by comparing translation and source text. Such annotators are not always available or affordable.

An alternative is a monolingual evaluation, when annotators compare the translation with one or multiple references instead of the source. For this type of evaluation, annotators do not need to understand the source language. Hiring such annotators is much cheaper than hiring bilingual annotators. Monolingual evaluation however creates a bias towards the reference. A good translation might get a low score because it is not similar to the reference. If possible, source-based evaluation should be preferable.

Direct assessment

One of the popular methods used to evaluate Machine Translation systems is direct assessment (Graham et al. [2015]). It is used in the WMT Translation task.

WMT Translation task is a competition to create the best translation system held by Workshop on Machine Translation. The WMT evaluation procedure has been relying primarily on direct assessment since 2017 (Bojar et al. [2016]).

By direct assessment procedure, the annotators score each sentence on a scale from 0 to 100 on how fluent and adequate it is. To compute a score for the system, the scores of each annotator are standardized, and the average is used as the system score.

Evaluation measuring post-editing effort

Another kind of evaluation procedure measures how much effort takes to post-edit a translation. Post-editing is a process when a person corrects the translation made by MT. The effort is usually measured by a metric that compares the translation with the post-edited translation. We can adapt reference-based automatic metrics for this purpose by feeding them the post-edited sentence instead of the reference translation. The widely used metric is HTER (Snover et al. [2006a]). HTER measures the number of post-editing operations needed to transform a translation into a post-edited translation divided by sentence length.

1.3.2 Automatic evaluation

While in principle ideal, manual evaluation is expensive and is only used in certain cases. There are also metrics aimed at automatic estimation of Machine Translation quality. They score the translation by computing the similarity between the translation and the reference.

There are two kinds of such metrics: pretrained and string-based metrics. The string-based metrics work by computing lexical similarity between the MT output and one or multiple reference translations.

Pretrained metrics (also called neural or embedding-based) are built on embedding produced by neural networks. Pretrained metrics can use semantic information to compute similarity, which allows them to show better correlation with human judgment and make better ranking than string-based metrics (Kocmi et al. [2021], Freitag et al. [2021]).

There is no clear evidence that some embedding metrics significantly outperform others. That makes it difficult for researchers to decide which automatic metric to choose for their experiments. To achieve better results, it is recommended to use statistical tests and report results of multiple metrics, including both string-based and pretrained (Kocmi et al. [2021], Marie et al. [2021]).

BLEU (Papineni et al. [2002]) is the most widely used string-based metric. BLEU counts matching n-grams between candidate and reference translation. BLEU is sensitive to how the dataset is preprocessed. That may lead to the situation when results produced by two research teams for the same dataset are incomparable because they used different preprocessing. Post [2018] proposed SacreBLEU to solve this problem. SacreBLEU uses standardized language-dependent preprocessing rules and produces comparable results.

ChrF (Popović [2015]) is string-based metric which consistently works better than BLEU (Kocmi et al. [2021]) while being less widely used. It counts matching character n-grams instead of word n-grams.

COMET (Rei et al. [2020]) is a pretrained metric which is often recommended (Kocmi et al. [2021], Freitag et al. [2021]) as a robust metric with wide language support. COMET is built on top of XLMRoberta (XLM-R) (Conneau et al. [2019]). It is finetuned to predict human scores given source, target, and reference sentences.

Among others popular automatic metrics, there are BERTScore(Zhang et al. [2019]) and YiSi (Lo [2019]) build on top of multilanguage BERT and BLEURT (Sellam et al. [2020]) built on top of English BERT, and PRISM (Thompson and Post [2020]) built on top of MT.

1.4 Power of Machine Translation system

It is crucial for our thesis to define the term the power of Machine Translation; we will also say how “strong” an MT system is, with the same meaning. In this section, we explain how it is defined. We present related findings about the relation between model capacity and MT power.

For the purpose of the thesis, the power of an MT system is operationally defined as the MT quality on some test sets as judged by either human annotators or an automated metric.

It is common knowledge that for deep learning systems, we can improve the system performance by simply using more data for training or by using bigger architecture. Bansal et al. [2022], Gordon et al. [2021] and Ghorbani et al. [2021] give evidences this also applies to MT systems. These works study how the loss of the MT model changes with changing data size and model size.

Bansal et al. [2022] study relation between data size and model loss. Their experiments are completed on English → German translation task. They vary data size from 500K to 512m sentences and try modifications of Transformer with the varying number of encoder and decoder blocks. The left graph on Figure 1.2 shows how test log-perplexity changes for each Transformer setting depending on the number of sentences in the dataset. We can see the clear trend that log-perplexity falls with increasing the dataset size. This fall however starts to slow down beyond 10^7 of sentence pairs. They also studied how the result changes when adding noise to the data or when using back-translated data and found that the trend remains the same (see right graph in Figure 1.2).

The log-perplexity is the function used as a loss function in MT training. It is not the same as model performance because model performance is measured by metrics covered in the previous section. Figure 1.3 shows the relation between perplexity and BLEU scores that Bansal et al. [2022] measured in their train models. We can see that, in general, a lower perplexity value corresponds to a higher BLEU metric.

Gordon et al. [2021] study effect of model scaling and data scaling. They vary the dimension of the hidden state and the number of encoding and decoding layers. Figure 1.4 shows results that they measured for three language pairs on a model with varying capacity. Again, we can see a clear connection between data size and cross-entropy.

Ghorbani et al. [2021] only study the effect of model scaling. They work with much bigger models than Bansal et al. [2022] and Gordon et al. [2021]. They tested it on two datasets: in one dataset, annotators translated source sentences,

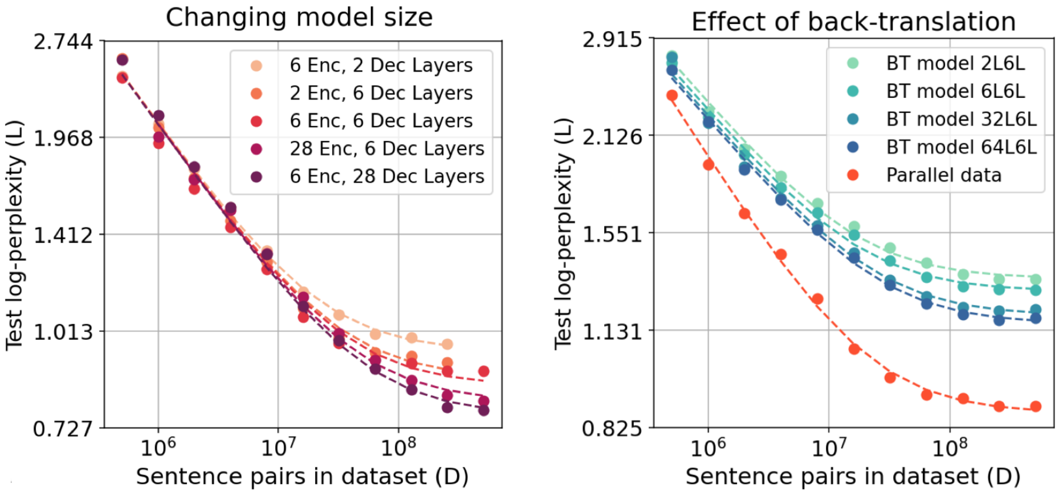

Figure 1.2: Effect of data size on model loss as measured by Bansal et al. [2022]

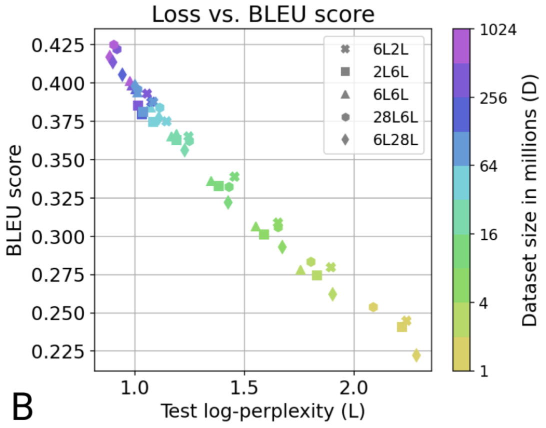

Figure 1.3: Relation between log-perpexity and BLEU scores measured by Bansal et al. [2022]

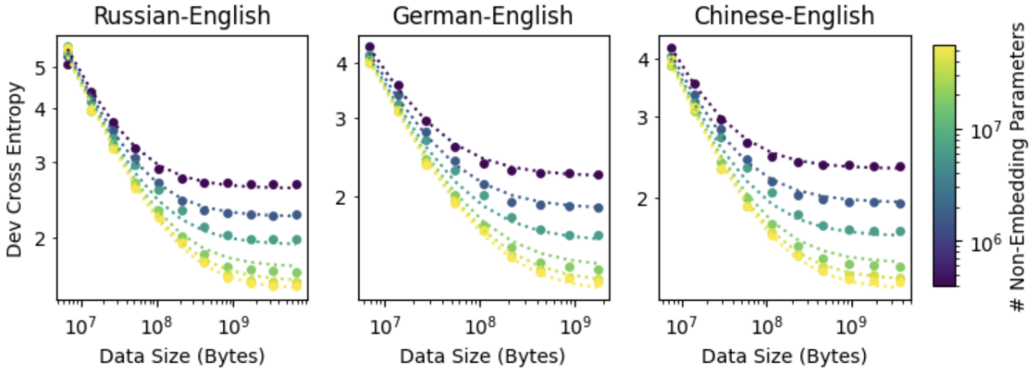

Figure 1.4: Effect of data size on model loss as measured by Gordon et al. [2021]

and in another backward dataset, annotators translated target sentences. Figure 1.5 shows how performance change when we scale a number of encoder and decoder blocks. The relation between loss and model size is similar to the relation we have seen with data amount.

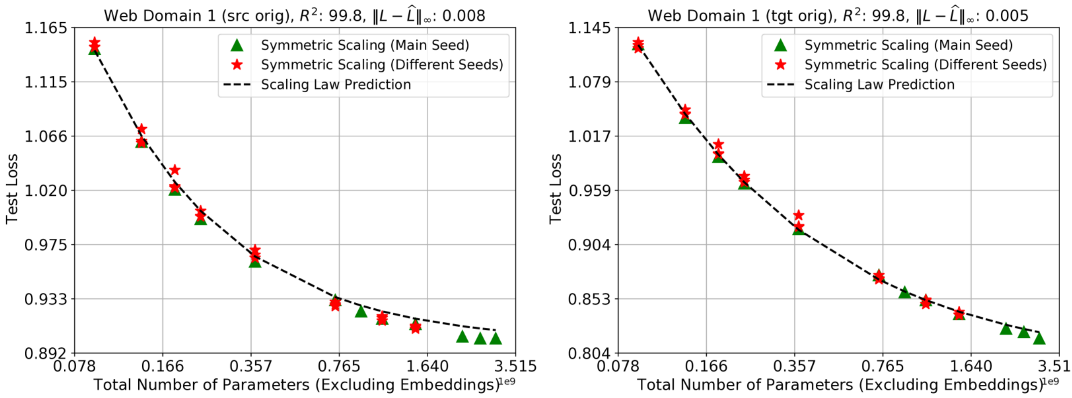

Figure 1.5: Effect of model size on model loss (Gordon et al. [2021])

Ghorbani et al. [2021] also measures how the model loss is connected with model performance. They measure model performance using the BLEU and BLEURT metrics. Figure 1.6 shows the relation between model loss and BLEU. That gives us mixed evidence since, for the right graphs, the relation is not obvious. BLEURT evaluation results mirror BLEU, so the issue is not caused by metric choice. The authors do not explain what caused the difference between left graphs and right graphs in Figure 1.6. If we sum up evidence gathered by Ghorbani et al. [2021] and Bansal et al. [2022], we can say that in most cases, lower model loss corresponds to higher model performance. Hence, the finding about the model loss scaling can be extended to the model performance.

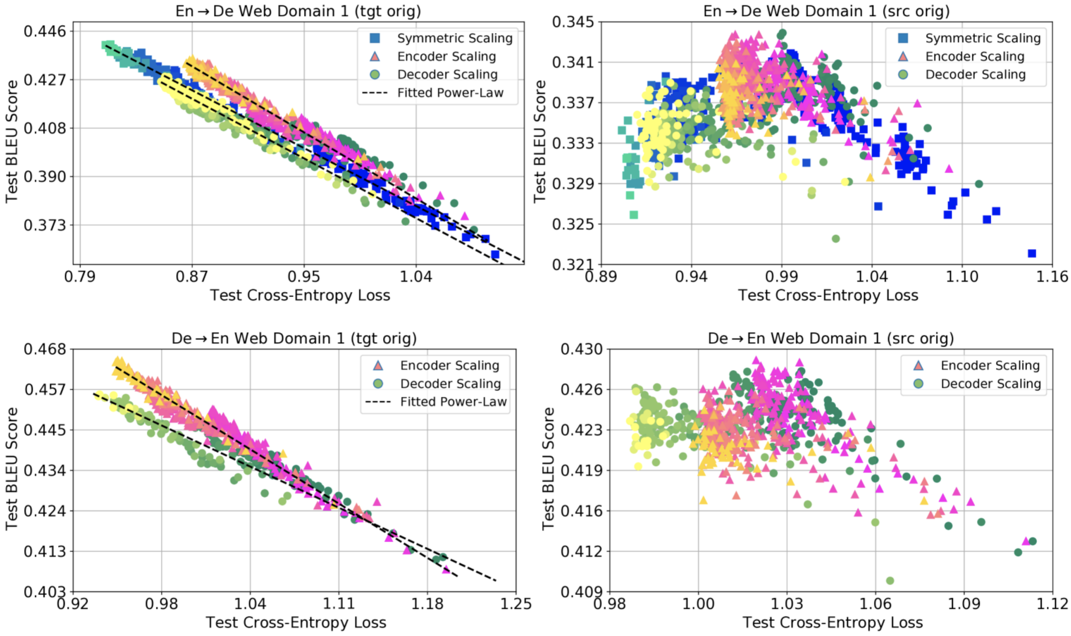

Figure 1.6: Relation between log-perpexity and BLEU scores measured by Ghorbani et al. [2021]

The studies we presented in this section study how the model size and dataset size affect the model power in diverse circumstances: they check it in multiple languages, use both authentic and back-translated data, train the model to translate backward, review the trend on models with small capacity and large capacity and with different proportion of encoder blocks to decoder blocks. Under all these settings, the trend is stable and clearly visible. The existence of this trend allows us to say that we can get the MT models of different power by adjusting the training data or model sizes. Moreover, we can expect the QE systems will exhibit the same trend, so we can also vary their power by changing capacity and training data amount.

2. Quality estimation

In this chapter, we introduce the concept of quality estimation. We describe the task, how it is connected with Machine Translation and what are existing solutions for this task.

Quality estimation (QE) is the task of predicting how good the translation is given the original and Machine Translation output (Blatz et al. [2004], Specia et al. [2009]). Quality estimation may work on word, sentence level, or document level. Quality estimation on word level is typically a classification task: each word is labeled as GOOD or BAD.

The goal of sentence-level QE is to predict scores assigned to the sentence by manual evaluation. This work is focused on sentence-level QE. Depending on which evaluation procedure is used to obtain data, there are different variations of the task: direct assignment QE, post-editing QE, and critical error QE.

Critical error QE is a classification task. Direct assessment QE and post-editing QE are regression tasks but have different data distributions. In the DA setting, the distribution has a mean equal to 0 since the labels of each annotator are standardized before computing the sentence score. Post-editing scores are computed by HTER, so scores are numbers between 0 and 1. The HTER itself might produce scores that are larger than 1 but they are usually clipped to 1.

2.1 Power of Quality Estimation system

We define the power of Quality Estimation system in a similar way as we define the power of Machine Translation system. The power of QE system depends on training data volumes and model hyperparameters. The more training data system consumes during training, the more powerful the system is. Model hyperparameters constitute another dimension that affects QE power. QE system power changes when we change the model capacity by increasing or decreasing the number of training parameters. Hyperparameters that define the model's architecture also contribute to power since some architectures tend to perform better than others even while having fewer training parameters. Besides that, using big pretrained masked language models such as Bert, XLM, and XLM-R also increases the power of the system.

2.2 Quality estimation vs. reference-free MT evaluation

QE does not have access to reference translation, which makes it different from MT evaluation which does not have such a restriction. Some MT metrics, mainly pretrained metrics, can also operate without a reference translation. The research area of reference-free MT metrics is very close to QE, but QE produces scores for a text segment, while MT evaluation produces scores for the MT systems. On the other hand, every QE system may, in principle, work as reference-free MT evaluation metrics by aggregating sentence scores to get the system score. That does not mean that a good QE system is as good in MT evaluation and vice versa. There is no guarantee that the approach that maximizes the system power in QE also maximizes its power as an MT metric.

2.3 Existing solutions

In this subsection, we talk about existing approaches to solving the task and frameworks implementing it. Nowadays, the task is usually solved by NN.

2.3.1 OpenKiwi

OpenKiwi (Kepler et al. [2019]) is a framework that implements QE systems. Currently it includes implementation of NuQE (Martins et al. [2016]) and Predictor-estimator (Kim et al. [2017a], Kim et al. [2017b]). OpenKiwi allows for training a new model from scratch and using model checkpoint to generate prediction and evaluation.

Predictor-estimator is the most widely used system from OpenKiwi. This architecture serves as a baseline system in WMT21 Quality Estimation task (Specia et al. [2021]) and actually belongs to the leading submissions (Chen et al. [2021], Zerva et al. [2021]) to this task.

The original model consists of a Predictor pretrained on parallel data and an Estimator trained together with a Predictor on QE data. The predictor is pretrained to predict the missing word in the translated sentence while supplied with the source sentence and the rest of the translated sentence. After the predictor is pretrained, the estimator and the predictor are trained together on QE data. The estimator takes word embeddings from the predictor and produces the prediction.

Using a big pretrained model such as BERT increases the performance of this architecture. OpenKiwi supports BERT, XLM, and XLM-R for this purpose. In this setting, the predictor produces sentence embedding of the concatenation of the source sentence and the translation. Base model of WMT21 and Chen et al. [2021], Zerva et al. [2021] use such model in their solution with multilingual XLM-R as a predictor. This approach also makes the model multilingual, which enables its usage for unseen languages.

2.3.2 Transquest

Transquest (Ranasinghe et al. [2020a]) is another framework that implements QE architectures. The framework offers two architectures: MonoTransQuest and SiameseTransQuest. Both architectures use XLM-R pretrained model to generate sentence embeddings. Ensemble of these architectures was used to create the submission to WMT2020 QE task (Specia et al. [2020]) by the Transquest team (Ranasinghe et al. [2020b]).

MonoTransQuest architecture takes as input a source and a translated sentence concatenated by [SEP] token. The input goes through XLM-R to obtain token-level embeddings. The first token embedding is fed to the softmax layer to compute the prediction. MonoTransQuest is a part of the leading submission (Wang et al. [2021]) to the WMT21 QE task.

In SiameseTransQuest architecture, embeddings for source and translation sentences are generated separately from two different XLM-R models. The QE score is assigned by computing the cosine similarity between two sentence embeddings.

2.3.3 DeepQuest

DeepQuest contains implementations of BI-RNN and POSTECH architectures. POSTECH is another implementation of the original Predictor-Estimator architecture. BI-RNN is a model that consists of two bidirectional RNNs. The source sentence goes through the first Bi-RNN, and the translation sentence goes through the second. The state vectors representing source and translation sentences are concatenated. The resulting score is computed with the application of the attention mechanism.

2.3.4 Glass-box features for QE

QE systems typically do not access the MT that translated source sentences. This is called the black box approach since the QE does not use information about how MT works internally. Fomicheva et al. [2020] proposed to extract metrics indicating how certain the MT is about the translation and use those as a quality estimation measure. These metrics are called glass-box metrics as a term opposing to black-box. Fomicheva et al. [2020] showed that such metrics are comparable with supervised approaches. Zerva et al. [2021] and Wang et al. [2021] also showed that combining them with input features boosts the performance of supervised models.

Fomicheva et al. [2020] use features from the probability distribution of the softmax prediction layer and the attention weight distribution. It is known that neural networks are bad in calibration. In other words, the probability that the model assigns to some words does not correspond to the actual frequency at which this word will appear in the valid translations. The Monte Carlo dropout (Gal and Ghahramani [2016]) helps to overcome the issue. The dropout is applied at inference time to generate multiple predictions. Aggregate metrics on these prediction catch uncertainty better than a single prediction without dropout. Wang et al. [2021] also use a similar approach that adds noise to training data instead of the model.

To give a better understanding of what features we are referring to, we will cover two of them that achieved the best results according to Fomicheva et al. [2020]: dropout translation probability and dropout translation similarity.

Translation probability (TP) is the average word log probability of the translation sequence. Dropout translation probability (D-TP) is an average of TPs computed with Monte-Carlo dropout:

$$TP = \frac{1}{|y|} \sum_{t=1}^{|y|} \log p(y_t | y_{i<t}, x, \theta)$$

$$D\text{-}TP = \frac{1}{N} \sum_{n=1}^{N} TP_{\hat{\theta}^n}$$

To compute lexical similarity, a set $\mathbb{H}$ of translations is generated by Monte-Carlo dropout. The Dropout Lexical Similarity (D-Lex-Sim) is computed as the average similarity score of translations pairwise comparisons. Meteor (Denkowski and Lavie [2014]) is used as similarity metric.

$$\textit{D-Lex-Sim} = \frac{2}{|\mathbb{H}|(|\mathbb{H}|-1)} \sum_{i=1}^{|\mathbb{H}|} \sum_{j=i+1}^{|\mathbb{H}|} sim(h_i, h_j)$$

2.3.5 PRISM

PRISM (Thompson and Post [2020]) is an MT evaluation metric. PRISM is trained as a Machine Translation system. The approach is similar to glass-box features: probability distribution of the softmax layer is used to estimate how certain the MT is about a prediction. The sentence score is assigned by computing translation probability, the same metric that was used by Fomicheva et al. [2020]. While Fomicheva et al. [2020] work with MT that generated translation, PRISM does not produce the translation but only scores translations generated by other systems.

PRISM exists in reference-based and reference-free versions. It is trained as a multi-language Machine Translation. The reference-based version makes the model translate from target language to target language. Here we consider only the results of a reference-free model as only a reference-free model can be used as a QE.

PRISM shows good performance in the MT evaluation task even when most of the models under evaluation are stronger MT than PRISM (Agrawal et al. [2021]. This can be considered as evidence that strong QE is not required to evaluate strong MT. We think this evidence is rather weak since the authors compare the model with other MT metrics, so this only shows that PRISM is strong against those metrics and not against the MT systems. This evidence could be stronger if the authors trained multiple PRISMs of varying MT power and showed that evaluation performance is not significantly higher in PRISMs of better MT performance. Moreover, this system has a bias towards translation similar to translation generated by itself. In experiments conducted by Agrawal et al. [2021], the MTs under evaluation are more similar to each other than to the PRISM model, so this bias is not observed.

2.4 Related findings on relation between MT power and QE power

In this section, we will describe related findings from papers featuring QE-related tasks that contain performance measures on models of different capacities (i.e. number of trainable parameters) or with different training data sizes. First subsection is dedicated to Agrawal et al. [2021] and Thompson and Post [2020] works. Second subsection is dedicated to Fomicheva et al. [2020] work.

2.4.1 PRISM

Agrawal et al. [2021] study different modifications of the original PRISM model, including the addition of new data and varying model sizes. They compare the original architecture used by Thompson and Post [2020] with two other architectures of different capacities: the Big model and the Massive model. Table 2.1 provides the details of the model configuration and average results on the test set measured by the Pearson correlation coefficient.

The results show a clear connection between model capacity and model performance. The difference in performance between the Massive model and PRISM is smaller than between the PRISM and the Big model. However, the order of parameter difference is almost the same: the PRISM model is roughly two times larger than the Big model, and the Massive model is two times larger than the PRISM. That suggests that the performance growth slows down with increasing the model size.

Name	**Params**	**Model configuration**				**Pearson**
		Layers	**Hidden**	**Heads**	**Model**	
Big	473M	6	8192	16	1024	0.808
Prism	900M	8	12288	20	1280	0.858
Massive	1.8B	8	16384	32	2048	0.883

Table 2.1: PRISM performance depending on model capacity (Agrawal et al. [2021])

To measure the effect of data size, Agrawal et al. [2021] extended the training data with a WMT15 parallel dataset. Original Prism model is trained on Prism-39 corpus consisting of 99.8M sentence pairs across 39 languages. WMT15 mainly consists of data featured in the WMT 2019 NewsTranslation Task (270M sentence pairs unequally distributed across 14 language pairs).

Data	**Pearson**
Prism-39	0.858
WMT-15	0.840
Prism-39 + WMT-15	0.867

Table 2.2: Effect of the training data.

Table 2.2 shows performance measures for each datasets. The performance gain is present but smaller than from increasing model capacity. For individual pairs, the relation between increased dataset size and the performance difference is not visible. The authors claim that it is caused by the data quality since the data have not been preprocessed. Hence there is not enough evidence to say that adding more data does not help with the training.

2.4.2 Comparison of unsupervised and supervised models

Fomicheva et al. [2020], as part of their evaluation, study how models behave for languages with different volumes of data available. Six language pairs were chosen for experiments to study models in different resource settings. We can consider it as studying how the size of the training data affects performance, taking into account that different languages bring noise to the measurements.

Table 2.3 shows the results. For each language pair, there is a dataset with parallel data for MT training and a dataset that contains annotated data for QE. We display the dataset statistics: the MT data size, the QE data size and the average quality score in QE training data that indicates the translation quality in test set.

	Low-resource		**Mid-resource**		**High-resource**	
	Si-En	**Ne-En**	**Et-En**	**Ro-En**	**En-De**	**En-Zh**
Size	646K	564K	880K	3.9M	23.7M	22.6M
QE Dataset Size	10K	10K	10K	10K	20K	20K
Average score	51.4	37.7	64.4	68.8	67.0	84.8
D-TP	0.460	0.558	0.642	0.693	0.259	0.321
D-Lex-Sim	0.513	0.600	0.612	0.669	0.172	0.313
PredEst	0.374	0.386	0.477	0.685	0.145	0.190
BERT-BiRNN	0.473	0.546	0.635	0.763	0.273	0.371

Table 2.3: Comparison of unsupervised and supervised methods measured for low-resource, mid-resource and high-resource language pairs Fomicheva et al. [2020] The numbers for the four evaluated models (D-TP, etc.) show Pearson correlation of the QE system and golden manual MT quality scores.

We analyze the results from Fomicheva et al. [2020] for two unsupervised methods described in the glass-box feature section and for two solutions used for comparison. We cannot use unsupervised methods to study the relation between MT power and QE power since those methods use one model for both MT and QE. However, we can find related findings in the performance of comparison methods. There are two comparison models: Predictor-Estimator and BERT-BiRNN. Predictor-Estimator is the model from the OpenKiwi toolkit. BERT-BiRNN is a BiRNN from the DeepQuest toolkit, which uses token-level representation generated by Bert.

We can see that all models perform best in mid-resource languages while performance in high-resource drastically falls. As the authors explain, high-resource MTs provide more smooth translation, leading to lower QE score variance. QE systems are struggling to capture slight variations in QE quality, leading to low scores.

PredEst model shows worse performance than the rest of the solutions. Its Predictor is trained on the same data as unsupervised solutions, and Predictor-Estimator is finetuned on the QE dataset, so it consumes the same amount of data as a better model. However, the model capacity is lower since it is an RNN-based model, which is outperformed by the Transformer. The low capacity is most likely the explanation for lower performance. Even if this model is worse on all language pairs, the gap is the biggest in the high resource settings. That might suggest that QE cannot fully use complex data if its capacity is low.

BERT Bi-RNN is a stronger model since it uses embedding from a more powerful BERT. It performs better than other models in the high-resource setting. In the low-resource setting, it is worse than unsupervised models. Predictor-Estimator also performs worse in the low-resource setting than in the middle-resource setting. That might mean that demand for data quality is higher for QE than for MT. In other words, low-quality data is not sufficient to train a good QE even if the QE model has a high capacity.

In the high-resource setting, the QE dataset is twice as large but it is difficult to tell if that affected the result given that high-resource languages have different data distribution than others.

3. Datasets and tools

This chapter covers our basic experiments. We describe the datasets we used for the experiments and how they were preprocessed to be used for the QE task. Then we describe what experiments we conducted, including details on how we train the MT and the QE. We present the results and explain what they mean for us and how they differ from what we expected to see.

We perform experiments to study the relationship between MT and QE. The goal is to vary QE training settings to find how those parameters affect system performance. The parameters constitute data size, data quality, and model capacity. The experiments are run on en → cs language pair. First, MT models are trained to translate from English to Czech. Those MT models translate the source sentence in the QE dataset. We get targets from computing the difference between the translation and reference sentences. In the following sections, we present this process in more detail.

3.1 Data

In this section, we present data used in the training and evaluation stages. The code that is used to preprocess datasets is available in Digital Attachment of this thesis in folder *src/data*.

3.1.1 Training dataset

The training data comes from CzEng 2.0 (Kocmi et al. [2020]). CzEng 2.0 is a sentence-parallel Czech-English corpus. CzEng 2.0 has authentic and synthetic (produced automatically by a first-pass MT system) data. In this work, we only use data from the authentic corpus. The authentic corpus consists of CzEng 1.6 extended with Europarl, News commentary, Wikititles, Commoncrawl, Paracrawl2, WikiMatrix, and Tilde MODEL Corpus. CzEng 1.6 is the previous version of the dataset, which, in turn, largely consists of movie subtitles, European legislation, and fiction. The authors cleaned the dataset by removing duplicates, noisy data from CzEng 1.6, and the data that very likely was non-Czech or non-English. The authentic CzEng 2.0 dataset has 61 million sentences. Since 2.0 is the only used version of CzEng, we skip mentioning the version and refer to the dataset as CzEng.

CzEng serves both as an MT training dataset and as a QE training dataset. Overall, we use around 20 million sentence pairs from the whole dataset. We divide it into two non-overlapping datasets, one for MT and one for QE. Each dataset, we further divide into train, validation, and test. The dataset statistics, such as the number of sentences, words, and distinct words, are presented in Table 3.1. In splitting, we consider document id, so sentences from one document do not go to different datasets.

MT dataset

	Train		Validation		Test	
	English	**Czech**	**English**	**Czech**	**English**	**Czech**
Sentences	10 000 000	10 000 000	4 885	4 885	10 000	10 000
Words	138 928 837	124 148 247	63 341	56 754	139 293	124 481
Distinct words	938 028	1 531 446	9 725	15 270	15 468	26 466

QE dataset

	Train		Validation		Test	
	English	**Czech**	**English**	**Czech**	**English**	**Czech**
Sentences	10 000 000	10 000 000	5 000	5 000	10 000	10 000
Words	138 991 405	124 209 751	63 924	56 731	137 975	123 047
Distinct words	937 565	1 532 088	9 553	14 942	15 413	26 082

Table 3.1: Number of sentences, words and distinct words for each language in each dataset.

3.1.2 Test datasets

The test sets from CzEng serve as evaluation datasets for MT and QE models. In addition, QE models are evaluated on other datasets, such as submissions to the WTM-2018 News Translation Task and IWSLT-2020 Non-Native Speech Translation Test Set.

Submissions to WMT18 News Translation Task

WMT18 dataset consists of submissions to WMT18 News Translation Task (Bojar et al. [2018]). WMT News Translation Task is a translation systems competition. Each competition participant submits a translation of the test set provided by competition organizers. The results are available for download on the WMT18 website.[1]

We use each participant submission as a separate dataset. The source sentences are the same for all datasets in the group. We provide their statistics in Table 3.2. The WMT18 dataset is more diverse than CzEng. If we compare the dataset stats with the QE dataset Validation English stats, we can see that WMT18 has longer sentences and more distinct words (12 548 vs. 9 553) while being smaller (55 920 vs. 63 924 words).

We named each dataset after the MT that created the submission. ONLINE-B, ONLINE-G, and ONLINE-A are made by organizers using online translation systems whose names are anonymized. CUNI-TRANSFORMER is MT submitted by Charles University (Popel [2018]), and UEDIN is MT submitted by the University of Edinburgh (Haddow et al. [2018]).

[1] https://www.statmt.org/wmt18/results.html

	English part of WMT18 test set
Sentences	2983
Words	55 920
Distinct words	12 548

Table 3.2: Number of sentences, words and distinct words in WMT18 test set (Bojar et al. [2018])

System	**Rank**	**Ave. %**	**Ave. z**
CUNI-Transformer	1	67.2	0.594
UEDIN	2	60.6	0.384
ONLINE-B	3	52.1	0.101
ONLINE-A	4	46.0	-0.115
ONLINE-G	5	42.0	-0.246

Table 3.3: Results of WMT-18 news translation task. Language pair en → cs (Bojar et al. [2018])

The submissions are evaluated using the DA procedure. The detailed explanation of this procedure is in Section 1.3.1. Table 3.3 shows the ranking of the systems, their DA scores, and z-normalized DA scores. We use this information to analyze how QEs perform with translations of varying quality. The higher quality data should be more challenging for QE, which, in turn, should lead to less precise predictions.

IWSLT-2020 Non-Native Speech Translation Test Set

The International Conference on Spoken Language Translation (IWSLT) is an annual scientific conference. Each year, it features competitions on the spoken language translation tasks that include translation from audio and text sources. We use the test set of the IWSLT-2020 Non-Native Speech Translation Task (Ansari et al. [2020]) as it is the only task in the last years that features en → cs translation. This test set was constructed by the ELITR project and is available on their GitHub page.[2]. For all datasets, we only use text transcriptions and reference translations.

IWSLT dataset consists of three parts: Antrecorp, Khan Academy, and SAO (auditing domain). Antrecorp and SAO preprocessing include segmentation into sentences, adding punctuation and casing, and then translation into Czech. We present dataset statistics in Table 3.4

The Antrecorp (Macháček et al. [2019]) consists of short business presentations made by students. The students are not native speakers of the English language, and therefore presentations have many grammatical errors. The personal information is removed from the dataset. To make the dataset look like a natural speech, we replace anonymization tags with real random data. For

[2] https://github.com/ELITR/elitr-testset/tree/master/documents/iwslt2020-nonnative-slt/testset

	Antrecorp		**Khan Academy**		**SAO**	
	English	**Czech**	**English**	**Czech**	**English**	**Czech**
Sentences	571	571	538	538	654	654
Words	7893	6505	4470	3466	13158	10818
Distinct words	1532	2104	871	1198	1897	3011

Table 3.4: Number of sentences, words and distinct words in IWSLT test sets

example, we replace a tag <NAME> with a random gender-neutral name, same for the source and the reference.

Khan Academy is a resource with educational videos. The dataset uses the subtitles from videos in the math domain. The dataset lacks proper segmentation into sentences as well as punctuation and casing. Because the dataset has a very narrow domain, it has 2-3 times fewer distinct words than other IWSLT datasets. Due to the absence of proper segmentation, the sentence length is also lower, and many sentences are segmented incorrectly.

SAO dataset is made for the Supreme Audit Office of the Czech Republic. It consists of presentations by officers of supreme audit institutions (SAI). SAO is an English dataset with speakers that are not native to English.

According to Table 3.4, the SAO dataset is the most diverse in the group. It has the longest sentences and the highest number of distinct words. That is happening because the SAO dataset contains texts made by professionals, while the Antrecorp consists of student presentations that have less experience in public speeches and therefore produce less sophisticated texts.

3.2 MT systems

As part of the experiments, we train MT systems. We control how much data is used to train the systems, so we get systems with varying power. We use those systems to translate the datasets and get machine translation that serves as input for a QE system.

When we translate the dataset with multiple MT systems with diverse power, we get QE datasets of varying quality. In other words, in a dataset generated by poor quality MT, QE learns to distinguish average translations from poor, while in a dataset generated by strong MT, QE learns to distinguish good translations from average. This way, we vary the quality of the data used to train the QE system. So, we have two ways to change QE input data: we can change the dataset size, i.e., cut the part of the whole dataset, or vary the dataset quality by using different MTs.

We train two MT systems. One system is trained on the whole MT dataset of 10 million sentences, and another is trained on a subset of 1 million sentences. In the following, we denote these systems as ‘1m’ and ‘10m’.

The trained MT systems are Transformers of base configuration in Marian implementation (Junczys-Dowmunt et al. [2018]). We use the default setting for Transformer provided in the Marian package and adjust parameter -w 6500, which changes the size of the preallocated workspace. We train the systems on two GeForce GTX 1080 Ti GPUs. Dataset preprocessing includes normalization, tokenization, and truecasing by Moses toolkit (Koehn et al. [2007]). Then the BPE tokenization is applied with 32 000 merge operations. The code that was used to train the models is available in Digital Attachment of this thesis in folder *src/models/mt_marian*.

We evaluated the systems using BLEU and chrF metrics to confirm that they correspond to our expectation of power. The evaluation results are available in Table 3.5. They show that, indeed, the system trained on 10 million sentences outperformed the system trained on 1 million sentences.

MT data size	**BLEU**	**chrF**
1m	31.07	0.5346
10m	36.02	0.5785

Table 3.5: Evaluation results of trained MT systems

3.3 QE dataset creation

QE datasets have a different structure than MT. One training sample for QE consists of a source sentence, a translated sentence, and a quality score which serves as a target value. We construct our training dataset from CzEng, which is a parallel corpus. The source sentences are taken from CzEng directly. The translation sentences are produced by the MT systems described in the previous section. The ordinary method to get the translation quality scores is to organize a human evaluation campaign. Given the size of the dataset and our available resources, it is impossible to acquire human annotations, so we create automatically generated targets.

We assign targets by computing a similarity between the translation and the reference translation taken from parallel CzEng data. As a similarity metric, we use TER (Snover et al. [2006b]), the artificial version of the HTER metric. We use TER since it is the industry standard for post-editing tasks. However, this metric has shortcomings, such as sensitivity to tokenization, limited range of possible values for short sentences, and missing ability to capture semantic information.

This metric computes the editing distance between translation and reference sentences. It is calculated as the number of edits needed to produce the reference sentence from the hypothesis divided by the reference length. As edits, TER considers insertions, deletions, shifts, and substitutions. The metric produces values between 0.0 and 1.0; if the value is bigger than 1.0, it is clipped to 1.0. The ideal translation has a score of 0.0 since it requires no edits. Low values indicate excellent translation.

Given our choice of TER, we are actually training the QE system to predict post-editing effort.

We apply the same procedure to generate validation and test sets from CzEng. Each MT generates separate sets. As a result, we have multiple validation and test sets. When we train a QE model, we pair the train set with the validation set

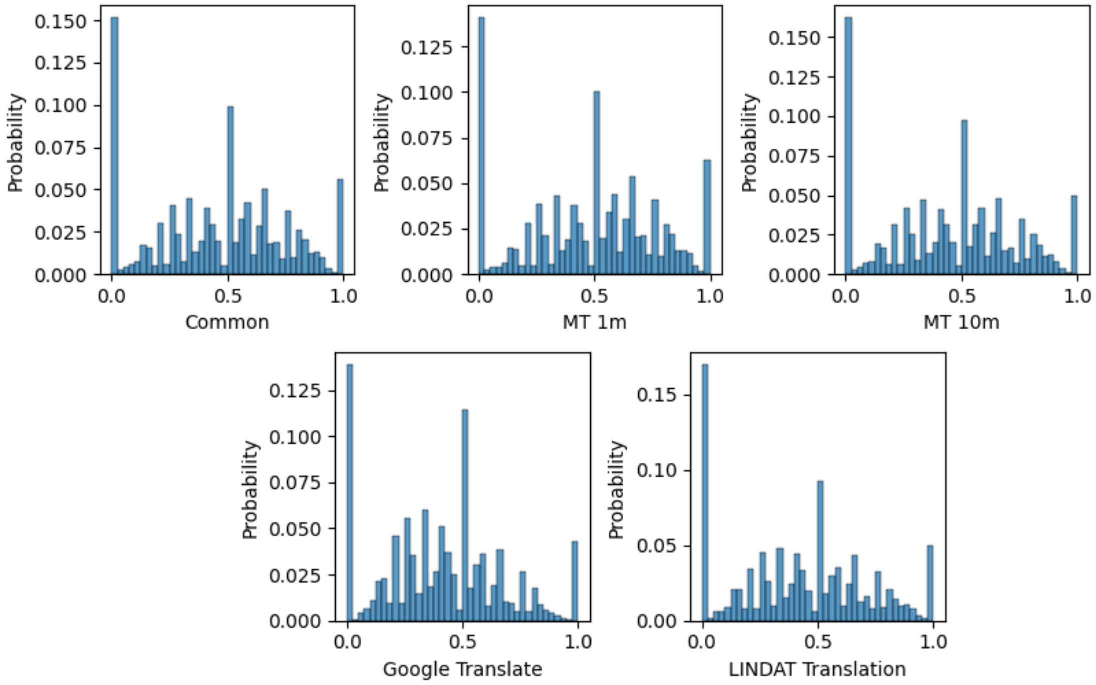

Figure 3.1: Targets distribution in CzEng dataset

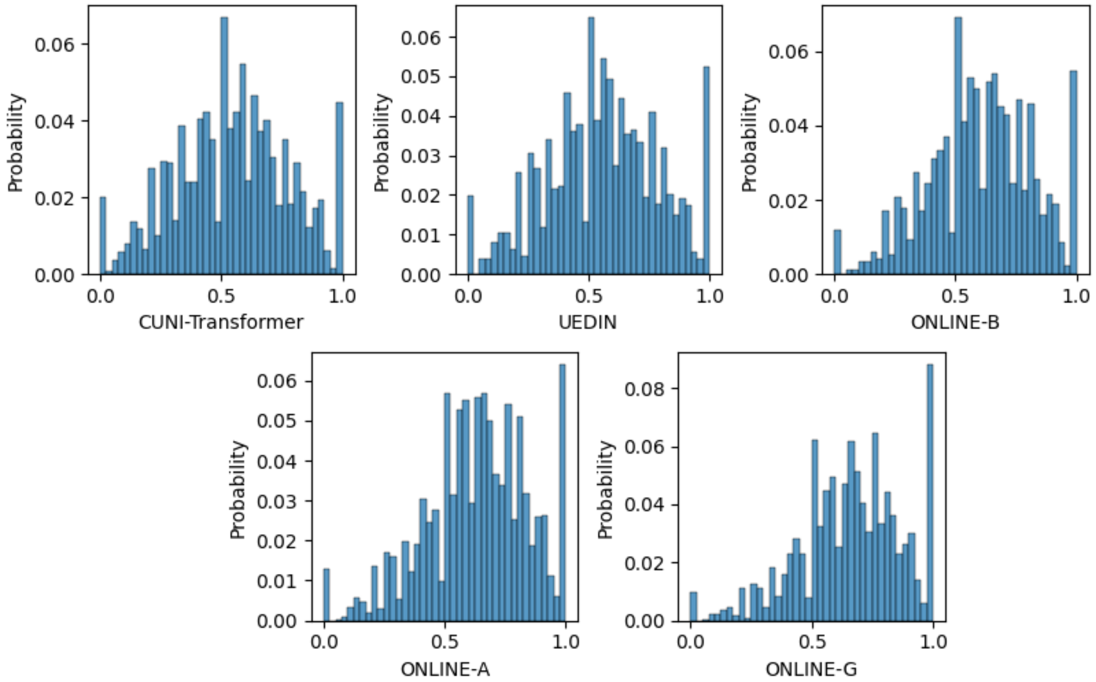

Figure 3.2: Targets distribution in WMT18 dataset

translated by the same MT system. During the evaluation, we evaluate models on each test set separately. In addition, we translate the CzEng test set by Google Translate and LINDAT Translation. Figure 3.2 shows targets distribution in the CzEng dataset. Each graph on the figure represents a separate MT system.

For WMT18 dataset, we use the translations provided in submissions. We translate IWSLT dataset with our MT and with Google Translate, and LINDAT Translation, where we try out sentence-level system (LINDAT Translation SL)

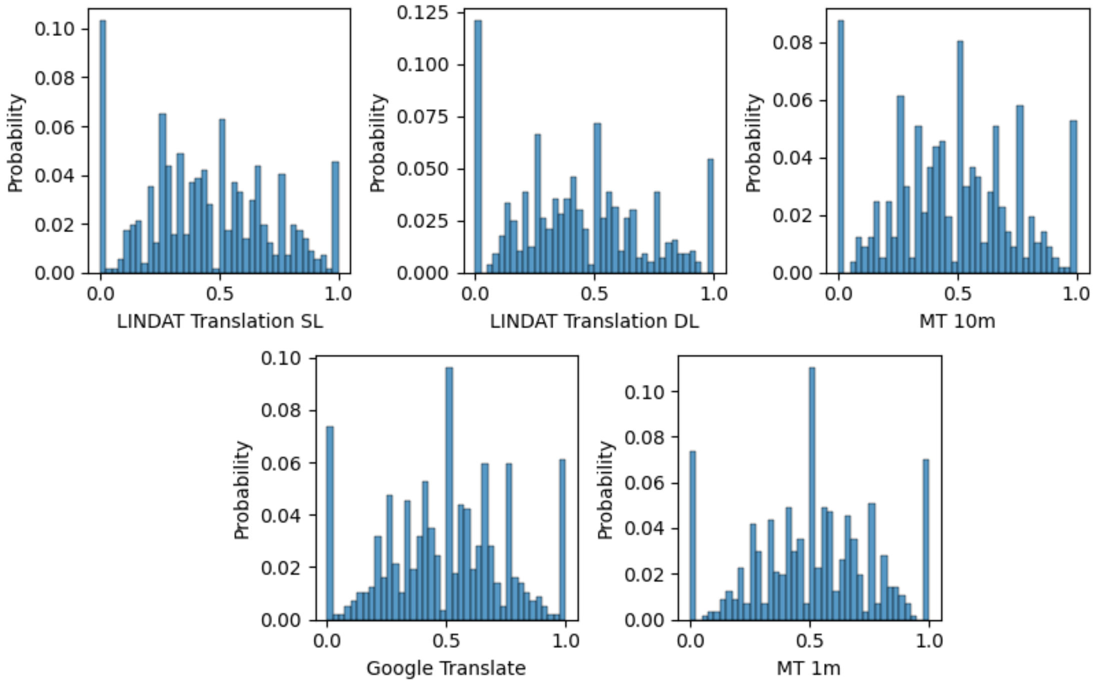

Figure 3.3: Targets distribution in IWSLT Antrecorp dataset

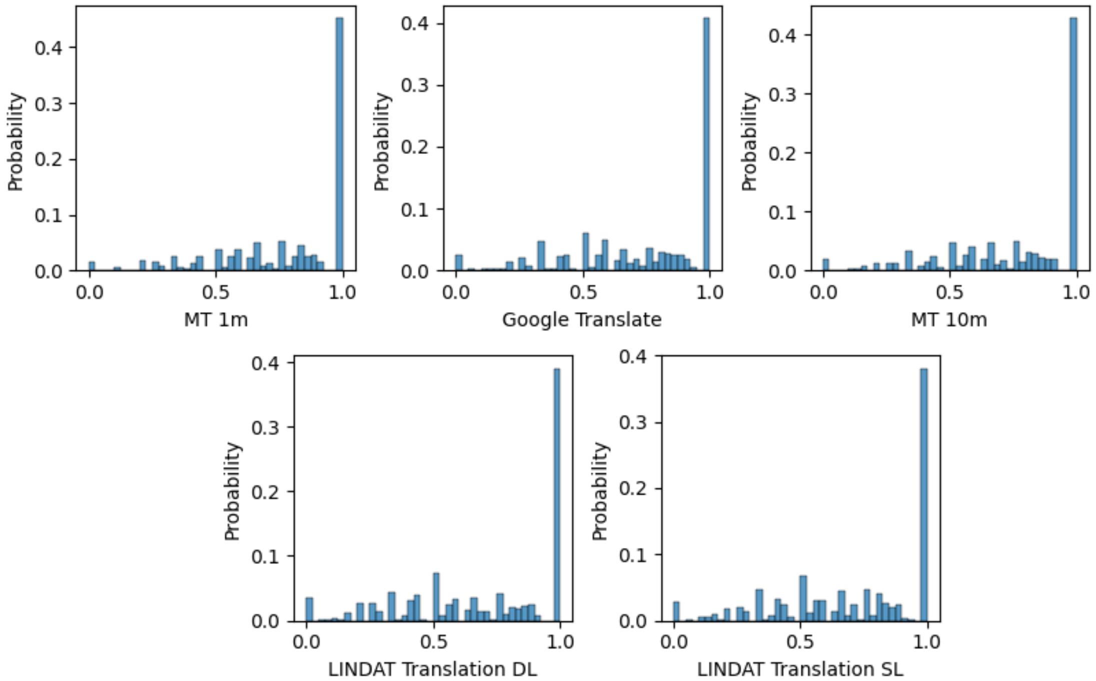

Figure 3.4: Targets distribution in IWSLT Khan Academy dataset

and document-level system (LINDAT Translation SL). Figures 3.2, 3.3, 3.4, 3.5 show targets distributions of these datasets.

CzEng and Antrecorp datasets have the biggest fraction of translations made without mistakes. As we mentioned earlier, they are less diverse than others and hence, less challenging for MT to translate. Moreover, the distributions within the dataset for different MTs look similar. The reason is that if the reference translation is short, there are limited options for target values. These values

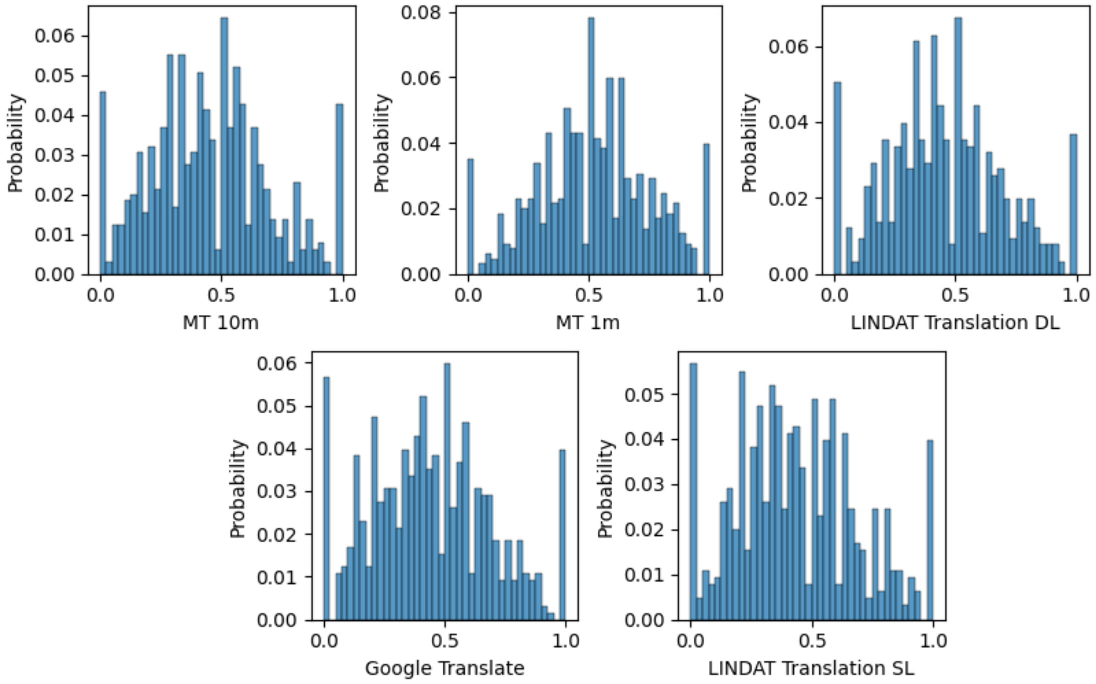

Figure 3.5: Targets distribution in IWSLT SAO dataset

create peaks on the graph, the same for all MTs. For example, for a reference of 4 words, the only possible values of the metric are 0, 0.25, 0.5, 0.75, and 1.0.

The Khan Academy dataset has a significant fraction of the data with entirely wrong translation. That is because the dataset is not properly segmented into sentences. Many segments have a context-specific translation that MT cannot capture due to the wrong segmentation, producing a more generic translation that turns out wrong.

3.4 QE systems

The goal of our experiments is to train multiple models and compare them to each other. We train the models using Bi-RNN architecture and Predictor-Estimator architecture. We measure the model performance of our model by computing the Pearson correlation between the predicted QE values and actual values for the whole dataset. The code that was used to train the models is available in Digital Attachment of this thesis in folders *src/models/qe_deepquest* and *src/models/qe_openkiwi*.

3.4.1 Bi-RNN

Bi-RNN architecture is implemented by DeepQuest. We covered this architecture in Section 2.3.3. We use this architecture since it allows us to easily change the model capacity by changing the size of the RNN layer and embedding layer. We use the config file *config-sentQEbRNN.py* as a base. We modify the parameters of the optimizer setting it to Adam with a learning rate of 0.003, and then change the batch size to 128. We preprocess input data in the same way as for MT training. We train the models on a single GeForce GTX 1080 GPU. Since our computational resources are limited, we train the models for no longer than two days. Many models neither reach a plateau nor start overfitting during this time. To make runs comparable, we limit the number of update steps, so models sharing similar settings have the same number of update steps.

3.4.2 PredEst

The second architecture is the Predictor-Estimator architecture (PredEst). We use it in OpenKiwi implementation covered in Section 2.3.1. As a predictor, we chose the XLM-R model. We selected this architecture since it is more powerful than OpenKiwi as it is based on pretrained models. Using this model allows us to compare performance on models of different power.

The training is based on config file *config/xlmroberta.yaml*. As we only implement sentence-level QE, we remove word-level related parts of the model from config. We adjust the learning rate to 5e-6, use 1000 warm-up steps and unfreeze the model after 2000 steps. We use a batch size of 4 with *gradient_accumulation_steps* equal to 4 to make the data fit into memory. We validate the model every 25 thousand sentences and stop the training if the Pearson correlation does not increase 25 times in a row. We train on the same hardware with the same limitations and data preprocessing as Bi-RNN model.

4. Experiments

4.1 Basic experiments

The main idea of our basic experiments is to train many QE systems and compare them to each other. The systems differ in settings that contribute to QE power. Among those, we consider model capacity, training data size, and training data quality. Each parameter could be seen as a separate experiment dimension. We take multiple points across dimensions and for each point combination, we train a QE model.

With each dimension we add, the number of possible experiments increases exponentially. That is why we avoid to check more parameters. We freeze all parameters that are not connected to system power using the same values for all our systems, so nothing else affects the results. One of such parameters is the language pair. Since it doesn't affect QE power, we're focusing only on one language pair, that being en → cs translation.

We consider 3 dimensions with two points in each dimension which results in 8 QE models. The dimensions are:

1. QE training data translation quality, measured by MT training size.
2. QE training data amount.
3. QE model capacity.
    - smaller model with hidden size 256 and embedding size 512
    - bigger model with hidden size 512 and embedding size 1024

We start by presenting our results on the test set from CzEng. Figure 4.1 shows evaluation results on CzEng translated by our MTs. Rows in the figure represent MTs used to translate dataset. MT 1m and MT 10m are named after the MT training data size: MT 1m and MT 10m, meaning 1 million and 10 million train sentence pairs. Common dataset combines data from MT 1m test set and MT 10m test set.

The columns represent the experiment dimensions. For each QE test dataset with CzEng source data, we have three graphs:

- On the first, the x-axis is data quality.
- On the second, the x-axis is the data amount.
- On the third, it is model capacity.

For example, the top-left graph in Figure 4.1 shows Pearson correlations of our QE predictions to the TER of MT output and the reference) when the test set translation came from both the weak and strong MT systems. The x-axis on graph represents data quality. On x-axis, point "1m" represents models trained on low-quality data, i.e. data produced by weak MT trained on 1 million sentence pairs from MT training dataset."1m" represents models trained on high-quality data, i.e. data produced by strong MT trained on 10 million of sentence pairs. This graph shows the relationship between data quality and QE model performance.

While x-axis on graphs represent some experiment dimensions, the y-axis always represent the QE model performance, i.e. Pearson correlation of prediction produced by QE model and targets. Since the y-axis always represent the same measurement, y-values in each row are the same.

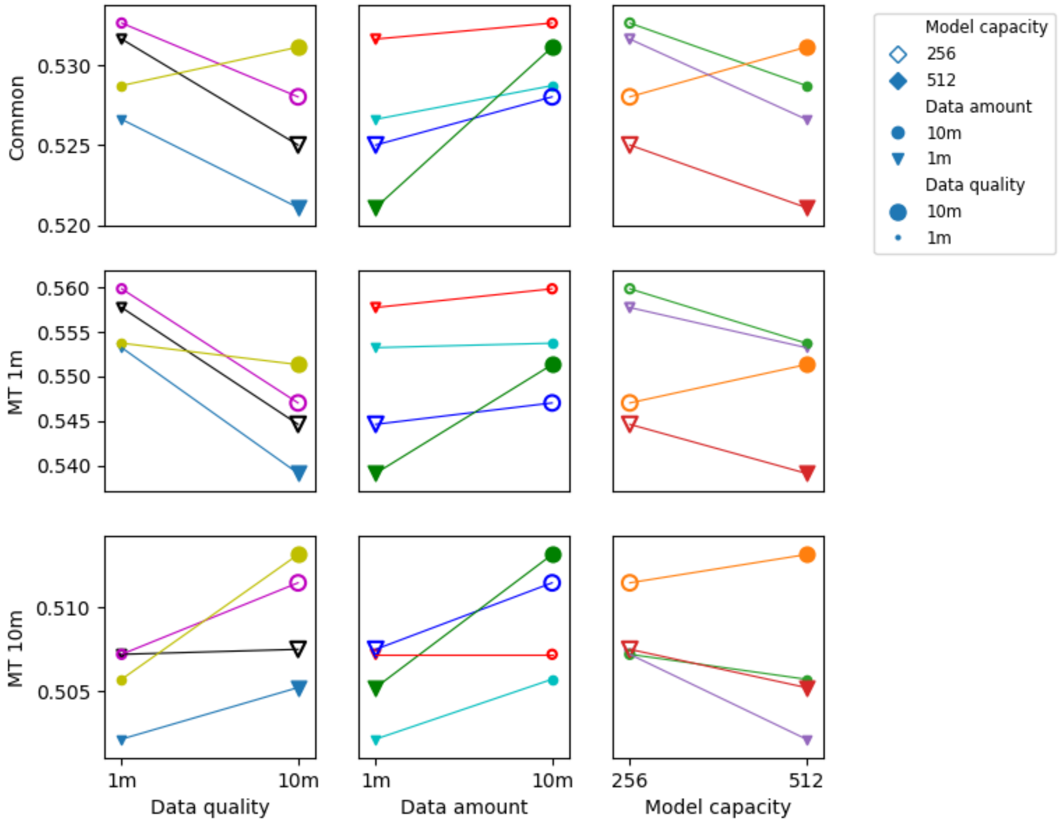

Figure 4.1: Pearson correlation with targets for CzEng dataset translated by our trained MTs. The top plots are for the test set that combines weak and strong MT outputs, the middle plots are for the test set containing the weak MT outputs (1m) and the bottom plots are for the test set containing the strong MT outputs (10m).

The points on the graph differ in form. The dot form is used to show the model setting. It is possible to decode which model the point represents from the dot shape. The encoding is the same for all points in the graphs. Whether the dot is filled with color or not corresponds to the model capacity. Circle and triangle dots indicate that the QE model is trained on 10m and 1m of sentence pairs, respectively. The data quality is represented by point size. The lines connect dots with the same configuration in all dimensions except the x-axis dimension.

For example, the top-left dot with a value of 0.5326 in the top-left graph represents the performance of the QE model trained on 10m sentence pairs from QE training dataset (the dot has a circle form) of low quality (the dot is smaller in size), and the model has the lower capacity (the dot is not filled with color). It is connected with the dot representing the model of the same capacity and data amount but trained on training data of higher quality.

Since visual representation varies for all three dimensions, one of the visual representations duplicates the x-axis. For example, on any left graph, all points with x=1m are small since the point size represents the data quality, which is also the x-axis dimension. In the middle graphs, all left points are triangular (low QE data amount) and all right points are circles (high QE data amount), etc.

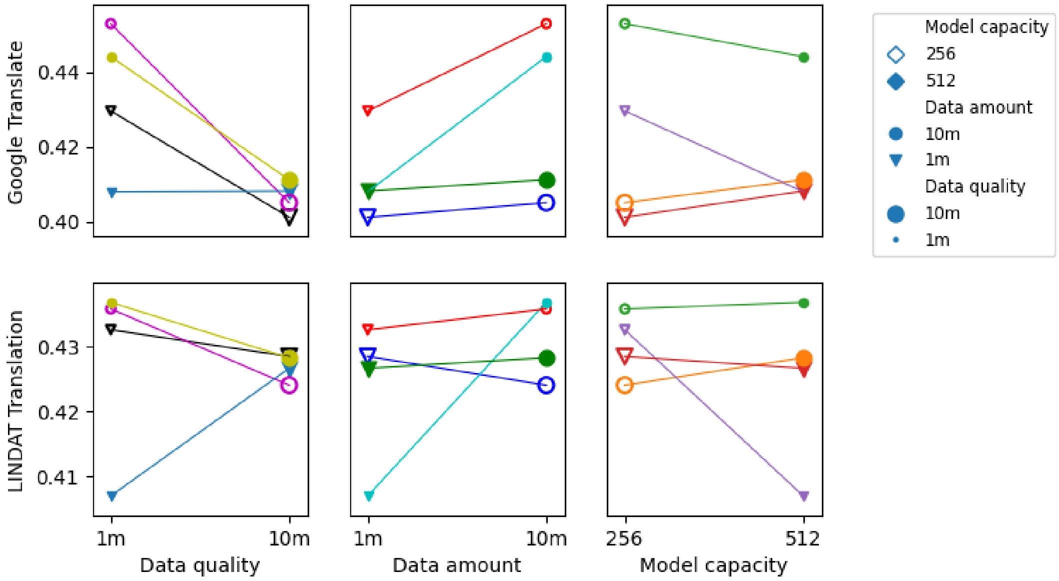

Figure 4.2: Pearson correlation with targets for CzEng test set. The top plots are for the test set translated by Google Translate system, and the bottom test set translated by LINDAT Translation

4.1.1 Evaluation results on MT 10m and MT 1m datasets

The first column of Figure 4.1 and Figure 4.2 shows relation between data quality and QE model size. We can see that on all translation systems (rows of plots) except MT 10m, generally, the models trained on high-quality data (10m) outperform those trained on low-quality data (1m).

The test sets MT 1m and MT 10m have a specific relation with the data quality dimension. High-quality QE models train set (right points on Data Quality graphs) and test set MT 10m (bottom row in Figure 4.1) share the MT that translated the data. That means that the QE models were trained on data translated by MT 10m system and test set was translated by the same system. On the MT 10m test set, high-quality QE models are evaluated on the data from the train distribution while the low data quality QE models are evaluated on the data with distribution shift from their trained data. In other words, the high-quality QE models have a handicap. That explains why the high data quality QE models outperform the low data quality, while on other datasets, the effect is the opposite.

The same relation holds for low data quality QE models and MT 1m test set. On MT 1m dataset, the low data quality models have the same bias. We can see that the absolute difference between the worst and the best model is higher on the MT 1m dataset (around 0.02) than on the MT 10m dataset (around 0.008), which is a direct outcome of this bias.

4.1.2 Data quality

On CzEng, the high data quality QE models have worse performance than low data quality QE models. This is an unexpected result. We expected the opposite: the QE exposed to good quality data should perform better on good quality test data, or in the worst case, there should be no difference.

This effect might be caused by low model power. The Transformer is a better translation model than RNN-based models. It is possible that the BiRNN model can only learn simple heuristics that help distinguish between a good and a bad translation. Data of low quality have more examples where such simple heuristics work, so it is easier for models to learn from low-quality data than from high quality.

The targets distribution also affects the result. The model learns the train targets distribution and therefore has a bias toward that distribution.

4.1.3 Data amount

There is also a mild effect from data amount (second column on all graphs). This can be seen in the middle column in all Figures except Khan Academy test results. The QE models trained on the larger amount of data performs better, but the difference is relatively small. This effect might be caused not only by the number of sentence pairs in the train set, but also be caused by longer training. The high data amount QE models were trained 1,5 times longer in a number of train steps so they could go through the whole dataset more times.

Given our limitation in computational resources, the only remaining choice is to train QE models on lower volumes of data. In this case, the QE model will start to overfit faster, so we can fully train the models within two days.

4.1.4 Model capacity

Increasing the model capacity does not cause any difference in QE model performance. We cannot improve the performance of this architecture by increasing the model size while keeping other settings the same. We can increase the model power in different ways; for example, use other architecture or use pretrained models such as Bert, XLM or XLM-R.

4.1.5 Results on IWSLT and WMT-18

We checked the model performance on IWSLT datasets and WMT18 dataset. The results are presented in Figures 4.3, 4.4, 4.5, 4.6. Figure 4.7 aggregates results of all QE models, so we compare performance between datasets. The Figure includes measurements for all MTs made by all QE models. The resulting range for a single dataset is wide since the results are also affected by other factors such as the MT quality or the variance of QE models.

Among IWSLT datasets, Antrecorp is the domain where the QE reaches the best performance. It contains simpler data than the SAO dataset, making it easier for the models to assess the quality. The QE is prone to the same mistakes as MT. For example, if the translation is incorrect due to the segmentation error,

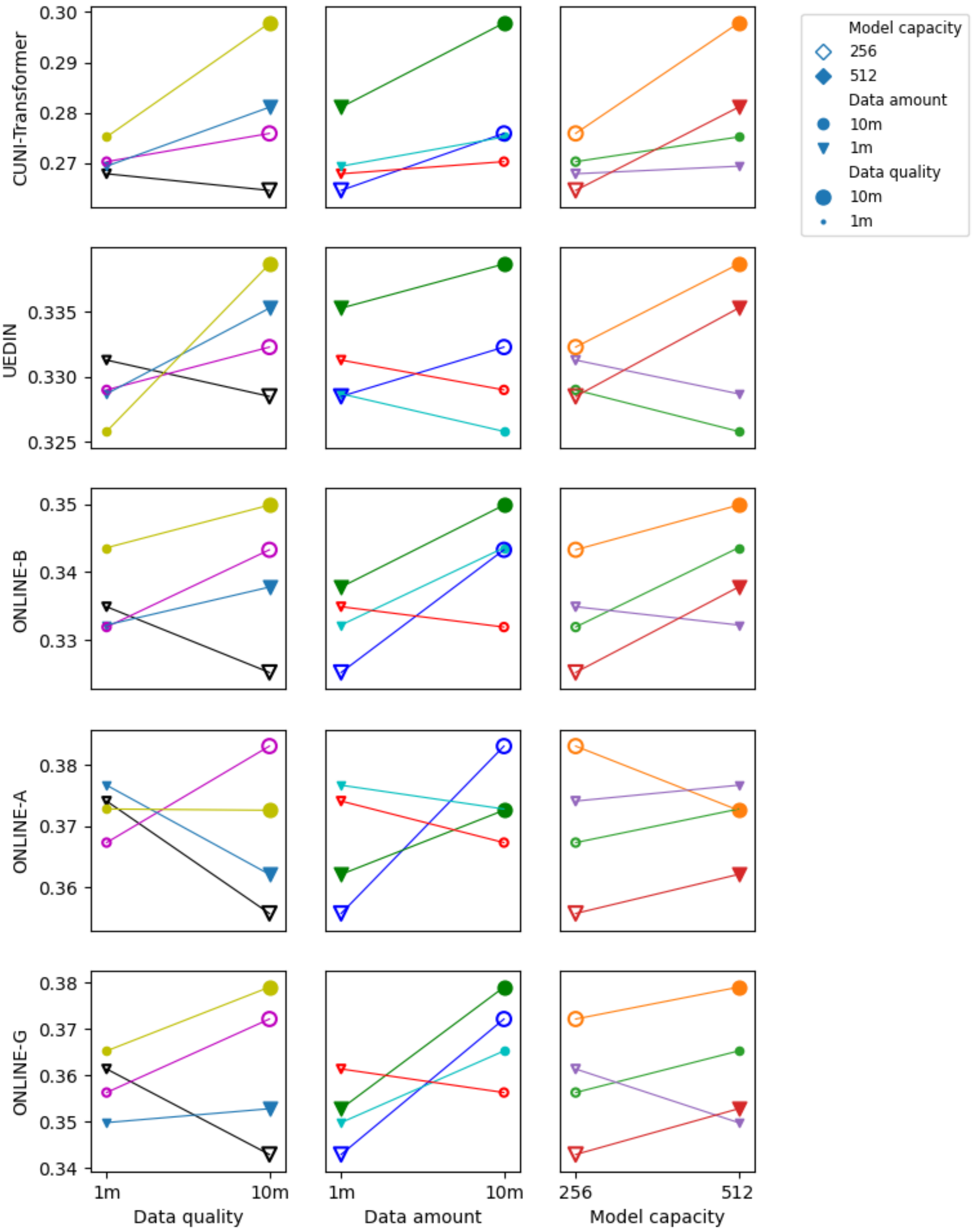

Figure 4.3: Pearson correlation with targets for test set WMT18 translated by WMT18 submission MTs

the QE labels it as correct. It explains why Khan Academy's performance is worse than Antrecorp's despite the worse data diversity.

On SAO domain, QE shows the worst results. Longer sentences require a more high-level understanding of sentence content which might be hard for QE. As a result, the QE tends to make more errors when working with long sentences. Moreover, it is more diverse than the other ISWLT group dataset, making the dataset more challenging for QE models.

On WMT18 domain, QE models shows results that are better than Antrecorp. WMT18 dataset is the most challenging from the data quality point of view, so

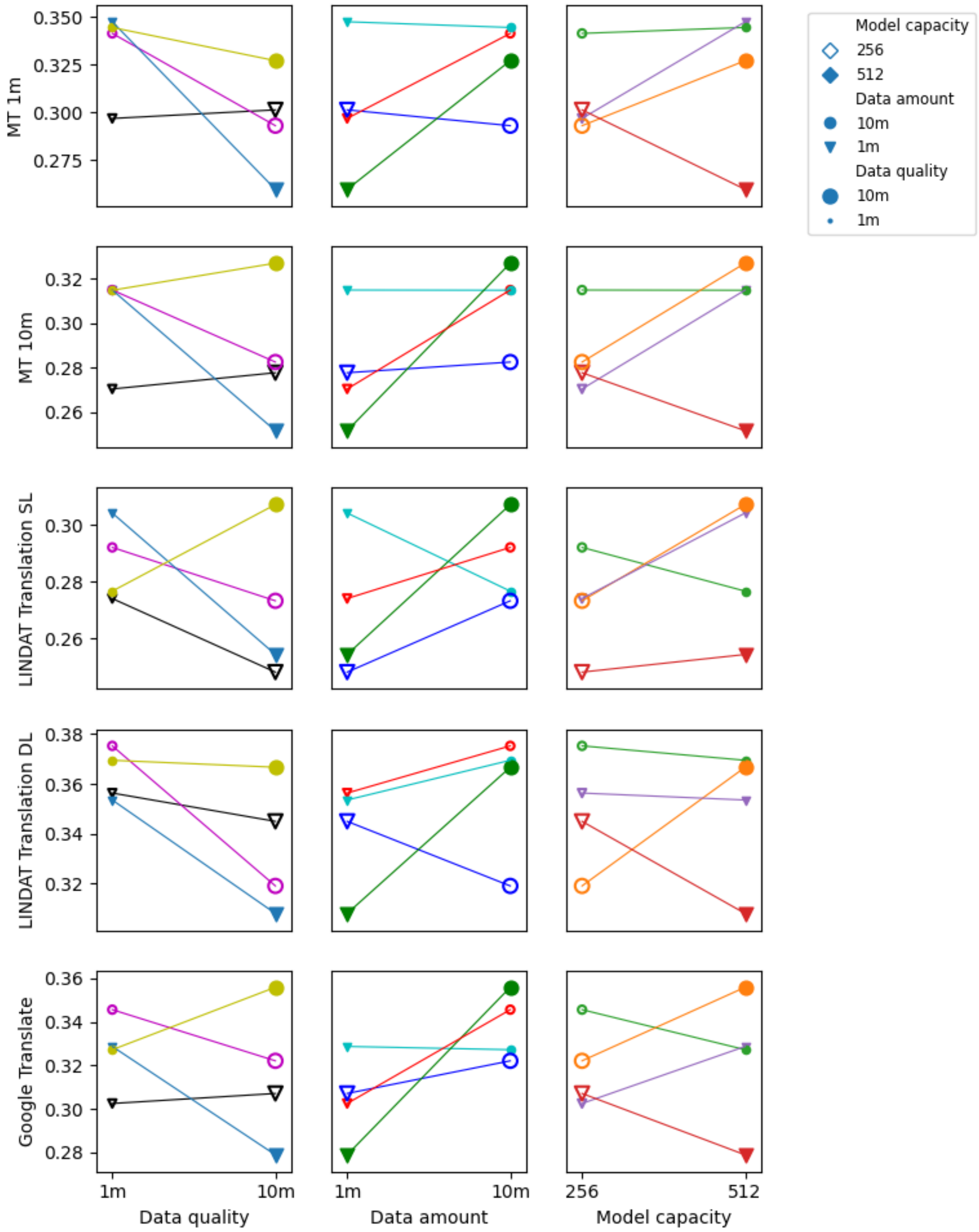

Figure 4.4: Pearson correlation with targets for test set Antrecorp from IWSLT translated by MT 1m, MT 10m, LINDAT Translation sentence-level model, LINDAT Translation document-level model and Google Translates

such results are somewhat unexpected. We should note that the WMT18 MTs are different but do not have lower power than the rest. As a result, the outcome cannot be attributed to the MT quality.

The WMT18 data domain is the closest to the training data. The sentence segmentation is correct, and the data do not have grammatical mistakes. We think that this is the reason for the good performance on WMT18. Given this result, we can say that the QE models are sensitive to the dataset quality and domain shift. The same factors that lead to a drop in MT performances also

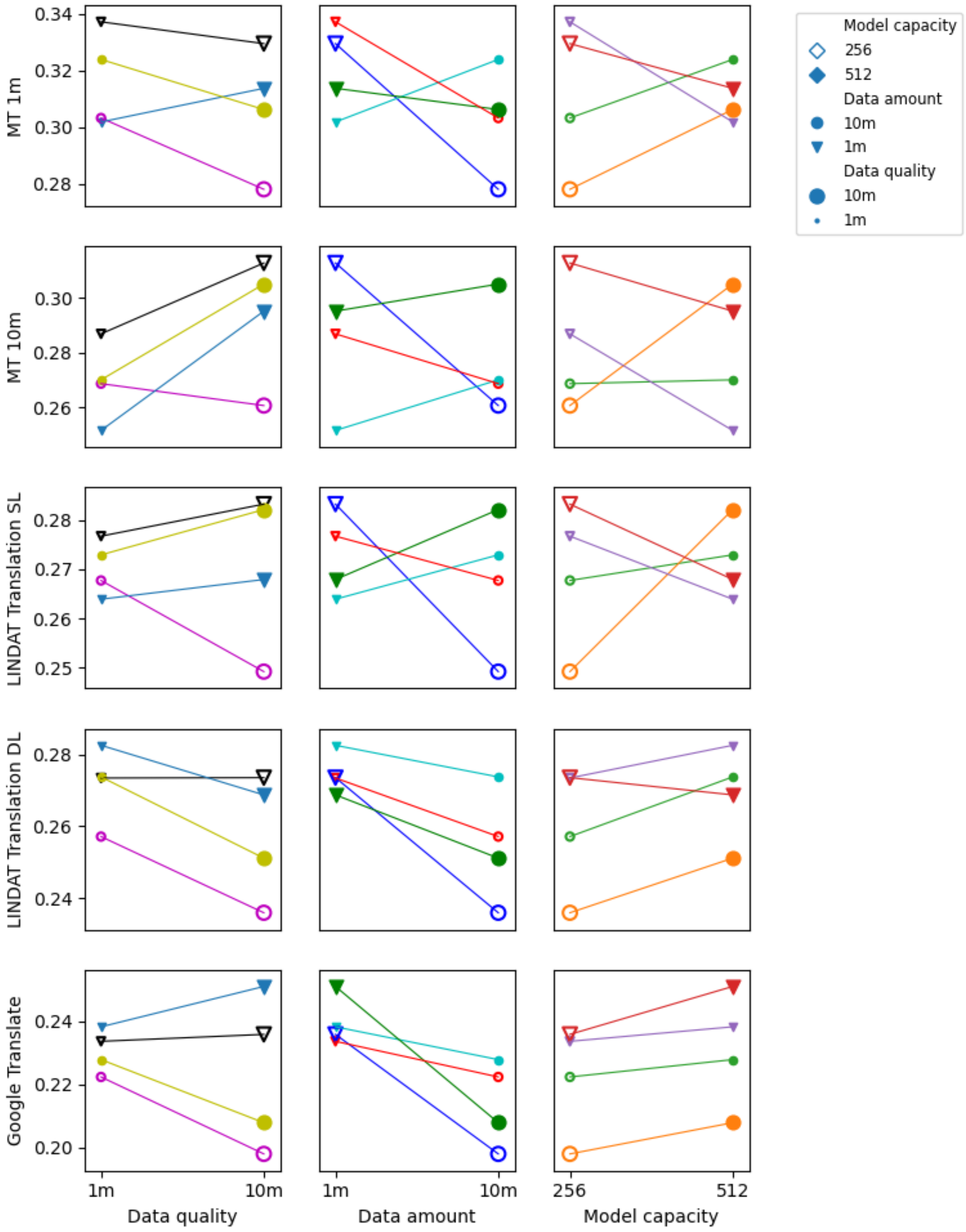

Figure 4.5: Pearson correlation with targets for test set Khan-academy from IWSLT translated by MT 1m, MT 10m, LINDAT Translation sentence-level model, LINDAT Translation document-level model and Google Translate

cause the QE performance drop.

4.1.6 Quality of translations in test datasets

QE models perform poorly on high-quality translation data. The more powerful MT translated the test set, the lower the correlation between predictions made by QE models and targets. We can see this effect on WMT-18 datasets, and when comparing MT 1m and MT 10m tests set on CzEng, Antrecorp, Khan-Academy

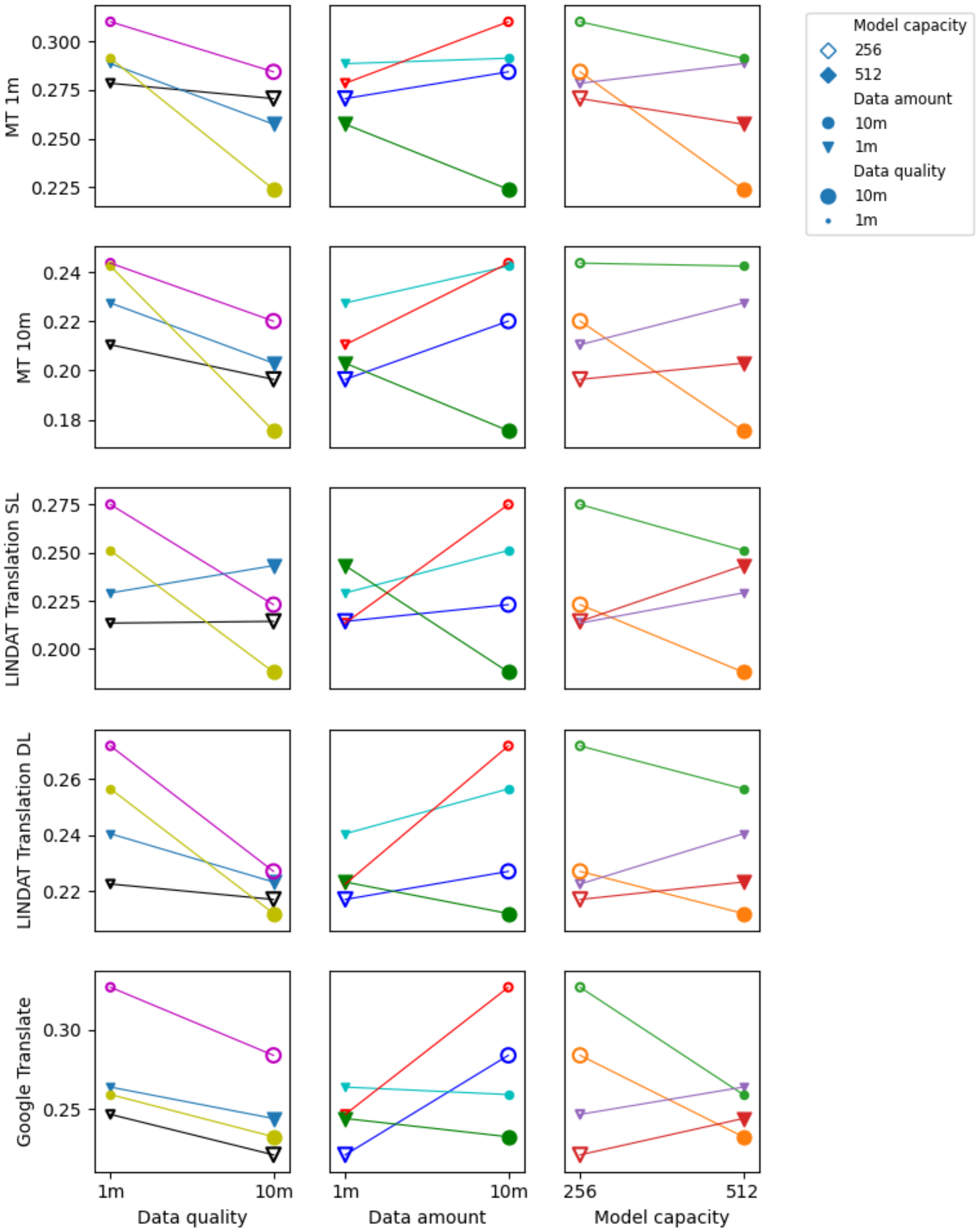

Figure 4.6: Pearson correlation with targets for test set SAO from IWSLT translated by MT 1m, MT 10m, LINDAT Translation sentence-level model, LINDAT Translation document-level model and Google Translate

and SAO.

In Figure 4.3 we have sorted rows according to MT scores assigned by DA (Table 3.3). The first row represents CUNI-Transformer that performed the best in WMT-18 News Translation Task. The second row is UEDIN, which took second place and so on. In Figure 4.3 CUNI-Transformer dataset has the lowest value of correlation between QE models predictions and targets (look at y-axis range). The UEDIN dataset has the second worse correlations values and so on. The trend is clear: lower correlation corresponds to the higher performance of MT.

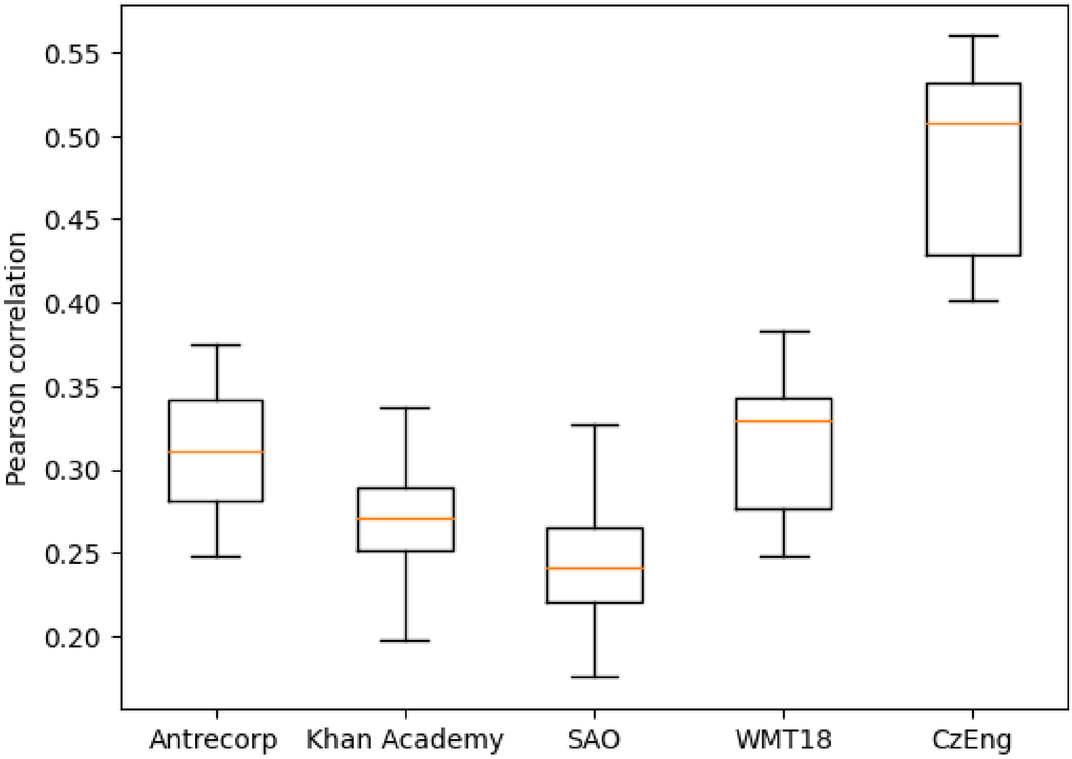

Figure 4.7: Range of Pearson correlations depending on the dataset

The only exception is ONLINE-A and ONLINE-G systems as their correlations highly overlap.

The same trend we observe when we compare results on the test set translated by our MT 1m and MT 10m. MT 10m test set contains data of higher quality since they are translated by stronger MT. We have such train sets for Czeng and IWSLT datasets. For all these datasets, the correlation on MT 1m test set is higher than on MT 10m test set.

Domain shift

CzEng test set evaluation results are much higher than in other datasets. That happens because the test data comes from the same distribution as the training data. Another bias comes from MT. We can compare results from CzEng and IWSLT to see the effect of these biases. Table 4.1 shows all four combinations and theirs effect on model performance. By 'Source bias' we mean that the test set source sentences come from the same distribution as sentences on which the model was trained and by 'Translation bias' that the MT model that generated translation is the same or similar model that generated the QE train set translation.

We compare the results to the IWSLT data with online MTs translations with no bias. On CzEng MT 1m and CzEng MT 10m, QE models have the highest scores due to both source and translation data bias. Even translated by online MTs, QE models evaluation results on CzEng are much higher than on other datasets. On the other hand, we see no difference between IWSLT test sets translated by MT 1m and MT 10m compared to test sets translated by Google

Dataset	MT	Source bias	Translation bias	Effect
CzEng	our MTs	Yes	Yes	Yes, large
CzEng	online MTs	Yes	No	Yes, medium
IWSLT	our MTs	No	Yes	No
IWSLT	online MTs	No	No	No

Table 4.1: Source and translation bias impact on model performance

Translate and LINDAT.

Therefore we can say that training on the data from evaluation distribution greatly impacts the model performance.

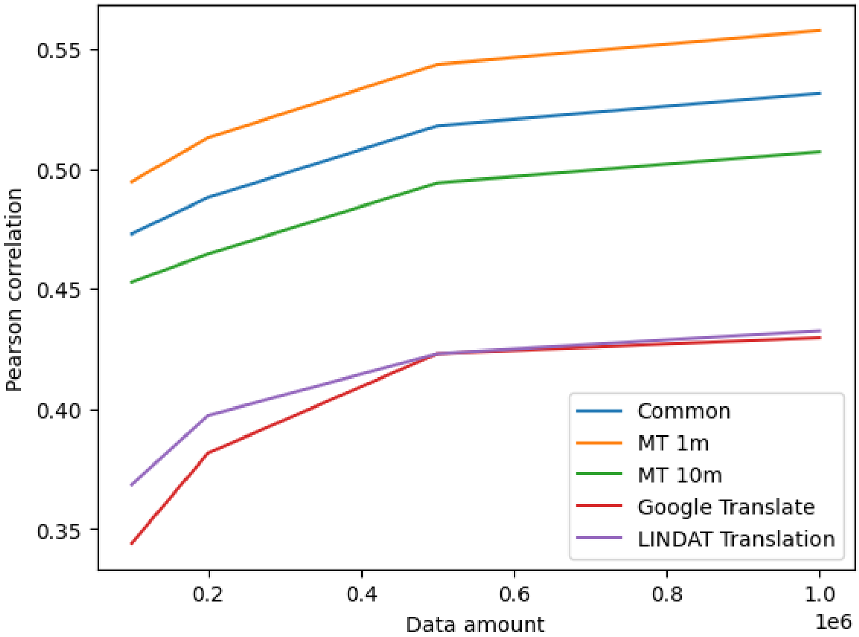

Figure 4.8: Pearson correlation with targets for CzEng dataset on models with varying data amount

4.2 Effect of dataset size

As we noticed in subsection 4.1.3, the models had not started to overfit when we trained them on 1 and 10 million sentence pairs. We trained the model on smaller data sizes to check this behavior. We fixed other settings that we vary in experiments and trained four models on 1e5, 2e5, 5e5, and 1e6 sentence pairs. We evaluated the model on all datasets.

The training time of the models differs depending on training data volumes. Models trained on 100 thousand sentence pairs quickly starts overfitting. 200 thousand models start overfitting before 500 thousand.

Figure 4.8 shows the evaluation result in CzEng datasets. We can see a strong relation between data amount and model performance. With more training data, the QE model achieves higher Pearson correlation scores.

QE models trained on 100k sentences quickly start overfitting. That raises the question of whether researchers in the field are using enough data as in the field, the datasets typically have a maximum of 20k sentences.

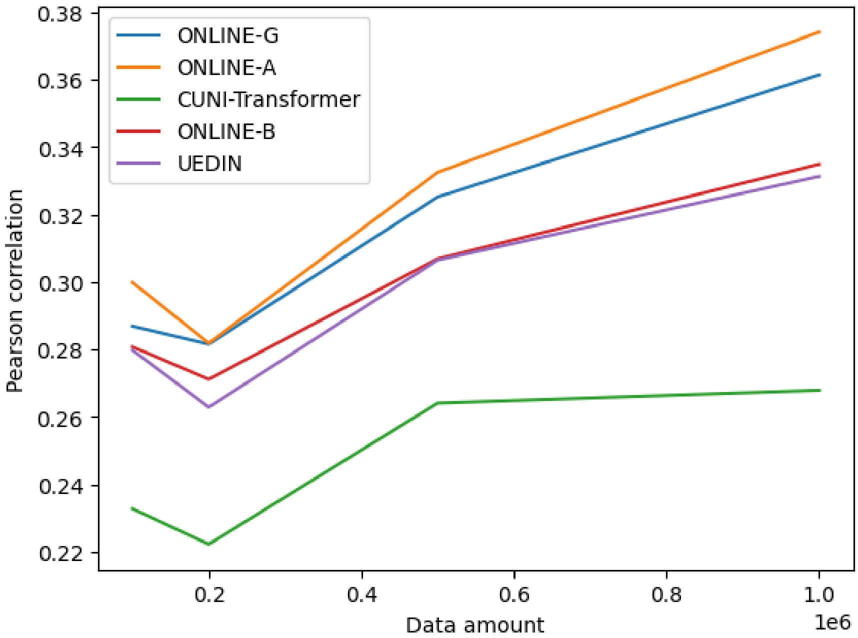

Figure 4.9: Pearson correlation with targets for WMT18 test set measured on models with training data amount that varies from 100 thousand to 1 million sentence pairs

Figures 4.9, 4.10, 4.11, 4.12 shows the evaluation on other test sets. It shows the same trend, but noisier. The model trained on 100 thousand sentence pairs is an outlier showing better performance than the 200 thousand model on many datasets. The effect is almost missing in SAO, which is also the dataset where QE has the worst result in absolute scores.

Even given the domain shift, the effect of the increasing dataset is present. If we have data from the same domain, we can train a good model on much lower volumes of data. This illustrates the fact that the 100 thousand sentence pairs model has higher performance on CzEng better than the 1 million model has on SAO.

4.3 Effect of dataset size with SOTA QE architecture

From the results of previous sections, we can see that increasing the model capacity does not affect the model's performance. Moreover, we observed that low-quality data performed better than those trained on high-quality. To investigate if this issue is caused by the low power of the model, we decided to train models of a higher power.

We trained the models with Predictor-Estimator architecture, whose implementation is provided in the OpenKiwi package. As a predictor, we used the base

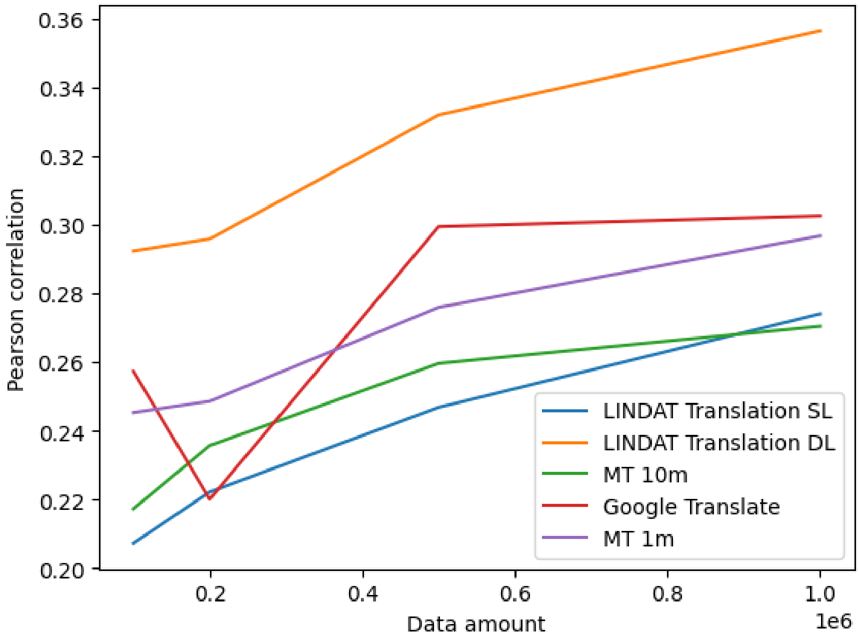

Figure 4.10: Pearson correlation with targets for Antrecorp test set measured on models with training data amount that varies from 100 thousand to 1 million sentence pairs

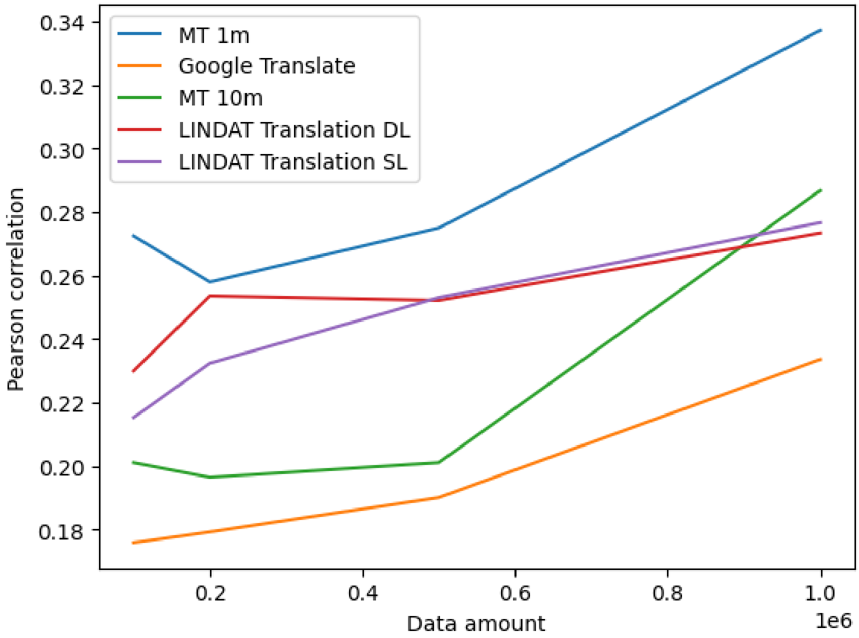

Figure 4.11: Pearson correlation with targets for Khan Academy test set measured on models with training data amount that varies from 100 thousand to 1 million sentence pairs

XLM-R model. Models whose Predictor is pretrained XLM-R model have much higher capacity than BiRNN model.

Since the predictor is a pretrained model, we cannot change its capacity. We adjust model power by varying the training data. We train the model on subsets

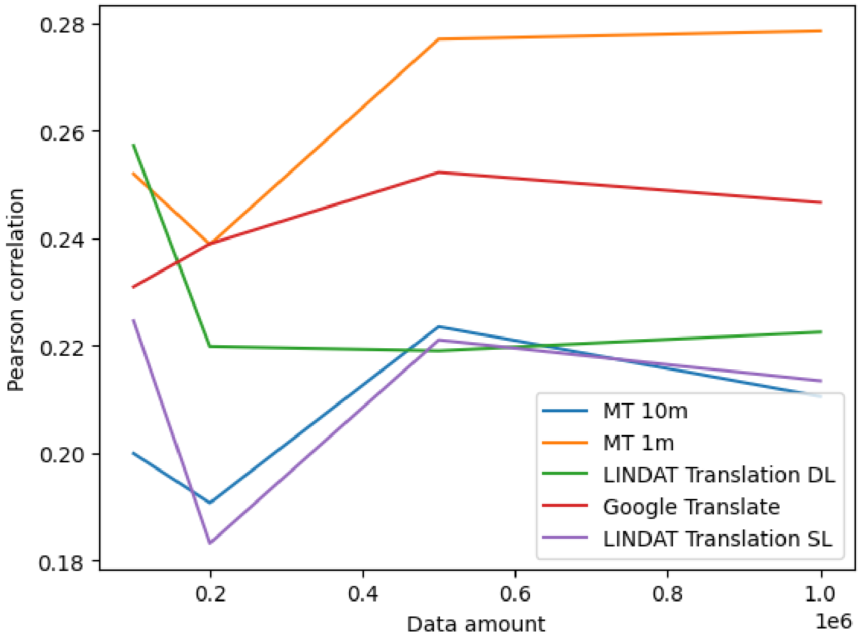

Figure 4.12: Pearson correlation with targets for SAO test set measured on models with training data amount that varies from 100 thousand to 1 million sentence pairs

of 100 thousand, 200 thousand, 500 thousand, and 1 million sentence pairs. We skip the 10 million sentence pairs dataset due to a long time of training. We vary the model quality in the same way as previously by using translation from MTs trained on 1m and 10m of sentence pairs. Figures 4.13, 4.14, 4.15, 4.16, 4.17, present the results of the experiments in the same manner as we presented the results of basic experiments.

4.3.1 Comparison of PredEst and BiRNN

Table 4.2 compares results from PredEst model to BiRNN model. We selected QE models that were trained on one million sentence pairs. We present measurements made on all test sets for different MT systems. As is expected for models of a higher power, PredEst models outperform BiRNN models.

On CzEng, the difference between PredEst and BiRNN is the smallest. With PredEst architecture, the model trained on 100 thousand sentence pairs perform on the same level as the BiRNN model trained on 1 million sentence pairs. The biggest difference is in Antrecorp and SAO datasets, which are from the auditing domain. For all datasets except CzEng, model power has the biggest impact on model performance from all factors we studied (data quality, data amount, model power). The PredEst generalizes better on out-of-domain data because PredEst is based on XLM-R pretrained on different domain data. The smallest difference is in Khan Academy dataset due to its skewed distribution.

For both model architectures, the highest correlation scores were achieved on CzEng test set. They also share the phenomenon when QE trained on low-quality MT data has higher correlation scores, which can be observed on all datasets but Khan Academy. However, a dataset with the lowest correlation for BiRBB was

Dataset	MT	MT 1m		MT 10m	
		PredEst	BiRNN	PredEst	BiRNN
CzEng	Common	0.586	0.531	0.579	0.525
	MT 1m	0.615	0.557	0.599	0.544
	MT 10m	0.555	0.507	0.560	0.507
	Google Translate	0.504	0.429	0.479	0.401
	LINDAT Translation	0.467	0.432	0.478	0.428
WMT18	ONLINE-G	0.491	0.361	0.457	0.342
	ONLINE-A	0.512	0.374	0.483	0.355
	CUNI-Transformer	0.404	0.267	0.377	0.264
	ONLINE-B	0.494	0.334	0.464	0.325
	UEDIN	0.462	0.331	0.434	0.328
Antrecorp	LINDAT Translation SL	0.390	0.274	0.376	0.248
	LINDAT Translation DL	0.397	0.356	0.391	0.344
	MT 10m	0.411	0.270	0.386	0.277
	Google Translate	0.459	0.302	0.445	0.307
	MT 1m	0.456	0.296	0.425	0.301
Khan Academy	MT 1m	0.413	0.337	0.405	0.329
	Google Translate	0.279	0.233	0.302	0.235
	MT 10m	0.287	0.286	0.342	0.312
	LINDAT Translation DL	0.269	0.273	0.301	0.273
	LINDAT Translation SL	0.289	0.276	0.320	0.283
SAO	MT 10m	0.338	0.210	0.305	0.196
	MT 1m	0.433	0.278	0.372	0.270
	LINDAT Translation DL	0.363	0.222	0.330	0.217
	Google Translate	0.366	0.246	0.307	0.221
	LINDAT Translation SL	0.384	0.213	0.348	0.214

Table 4.2: Comparison of PredEst and BiRNN models trained on one million of sentence pairs on datasets of different quality

SAO, while PredEst performs the worst on Khan Academy dataset.

4.3.2 Data quality

We expected different results in the data quality dimension because this type of models should have enough capacity to learn from data. Nevertheless, even with stronger PredEst architecture, models trained on low-quality data perform better than models trained on high-quality data. This effect is the most visible on the dataset whose domains are close to training data such as CzEng and WMT18, except for CzEng MT 10m subset due to the models' bias (see subsection 4.1.1). This means that the powerful MT is not needed to train good QE model. To investigate this further, we can train MT models of even lower power to find the data quality on which QE power has a peak performance.

We can also see that the gap between models of the same data amount differs

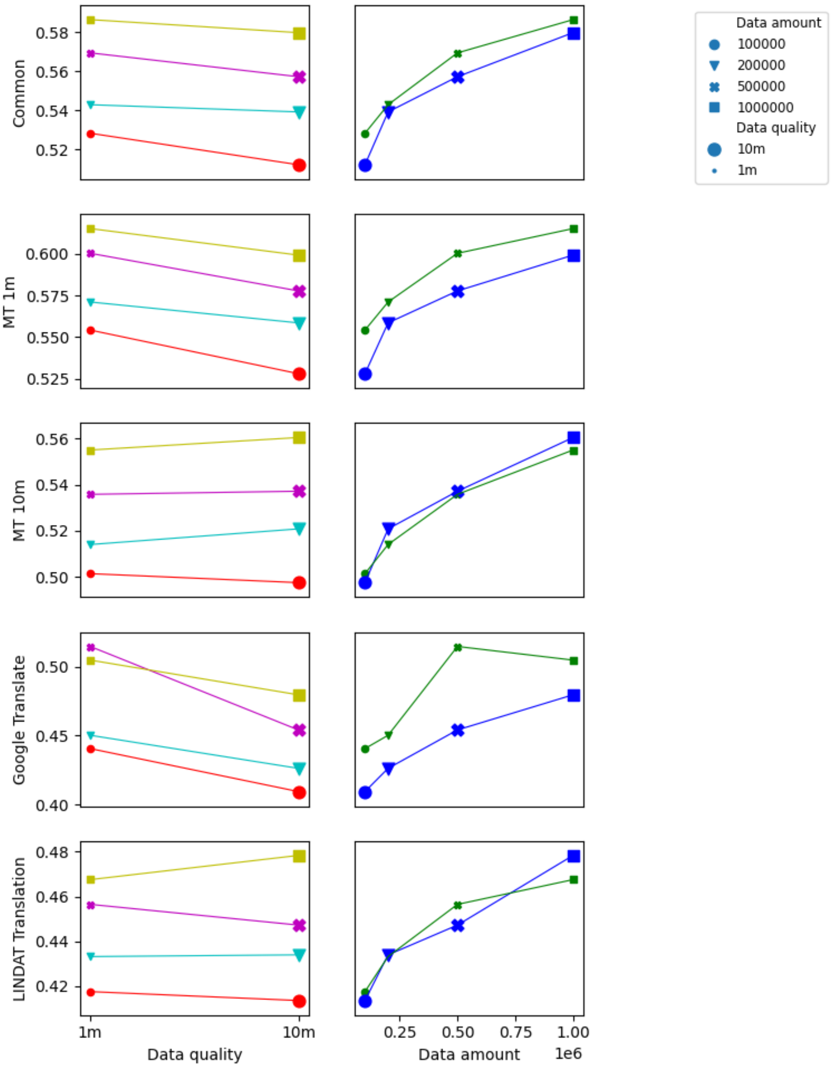

Figure 4.13: earson correlation between predictions generated by PredEst models with targets for CzEng test set translated by MT 1m, MT 10m, Google Translate and Lindat Translation sentence level. Common test set combines data from test set MT 1m and MT 10m

across subsets and datasets, but within one subset, it is mostly constant. For example, Figure 4.14 shows that on the CUNI-Transformer dataset, the gap between low-quality and high-quality data is almost absent. In contrast, low data quality models perform much better on the ONLINE-G dataset. We believe there is some similarity between datasets, so if datasets are similar, the model trained on a similar dataset will perform better than models trained on other datasets. The targets distribution might also cause such an effect.

In Figure 4.14, graphs are sorted according to their competition ranking. The better the WMT18 MT system, the closer the gap between low data quality and high data quality models. Better MT produces translations that are more challenging for QE to assess quality. The high data quality models have seen more of such data, which affects the result.

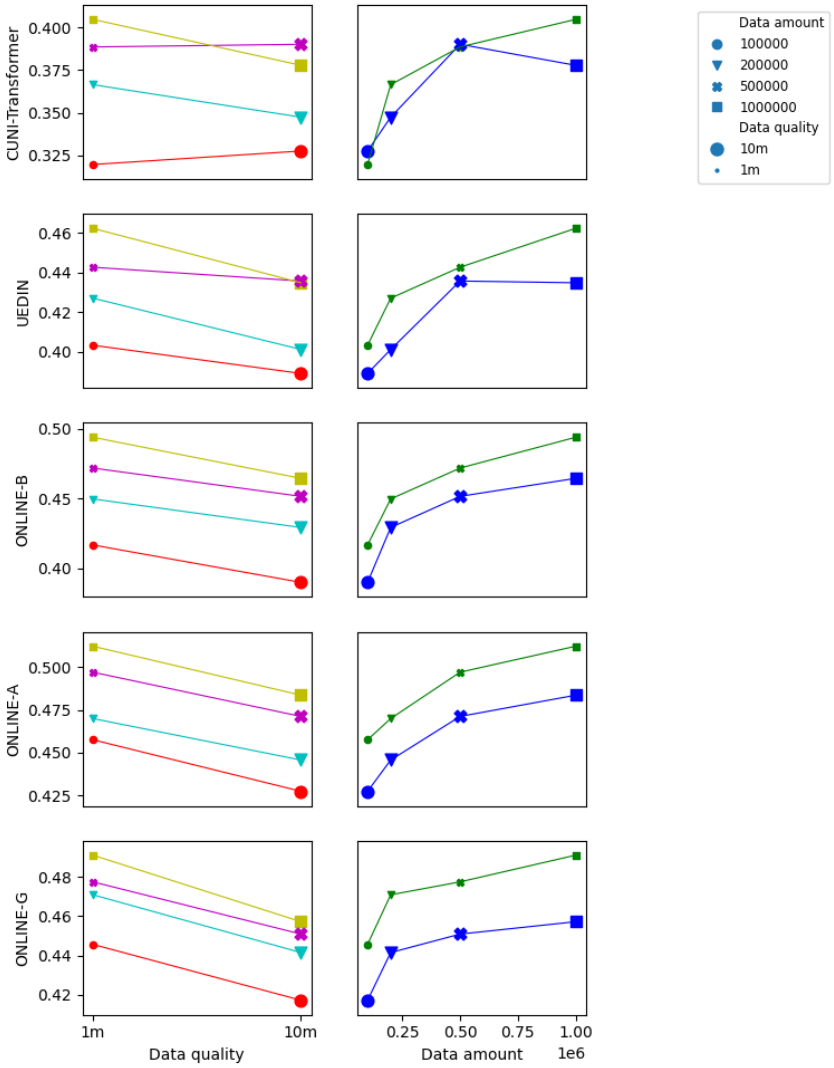

Figure 4.14: Pearson correlation between predictions generated by PredEst models with targets for WMT18 test set translated by WMT18 submission MTs.

4.3.3 Data amount

The data amount dimension of the model has a much higher variation than the data quality dimension. The models hugely benefit from adding more data.

This effect is visible on all datasets except for the Khan Academy dataset. However, on IWSLT datasets, the winning models are trained on 200-500 thousand of sentence pairs. When trained on large data, models tend to overfit to the dataset distribution. Hence, the ability to generalize to other domains decreases. This problem can be overcome by using the validation dataset from the testing distribution.

For the Khan Academy dataset (Figure 4.16), there is no effect of data size. This dataset has a skewed distribution with many context-dependent sentences. Giving more data does not make models more exposed to such data, so there is no performance boost.

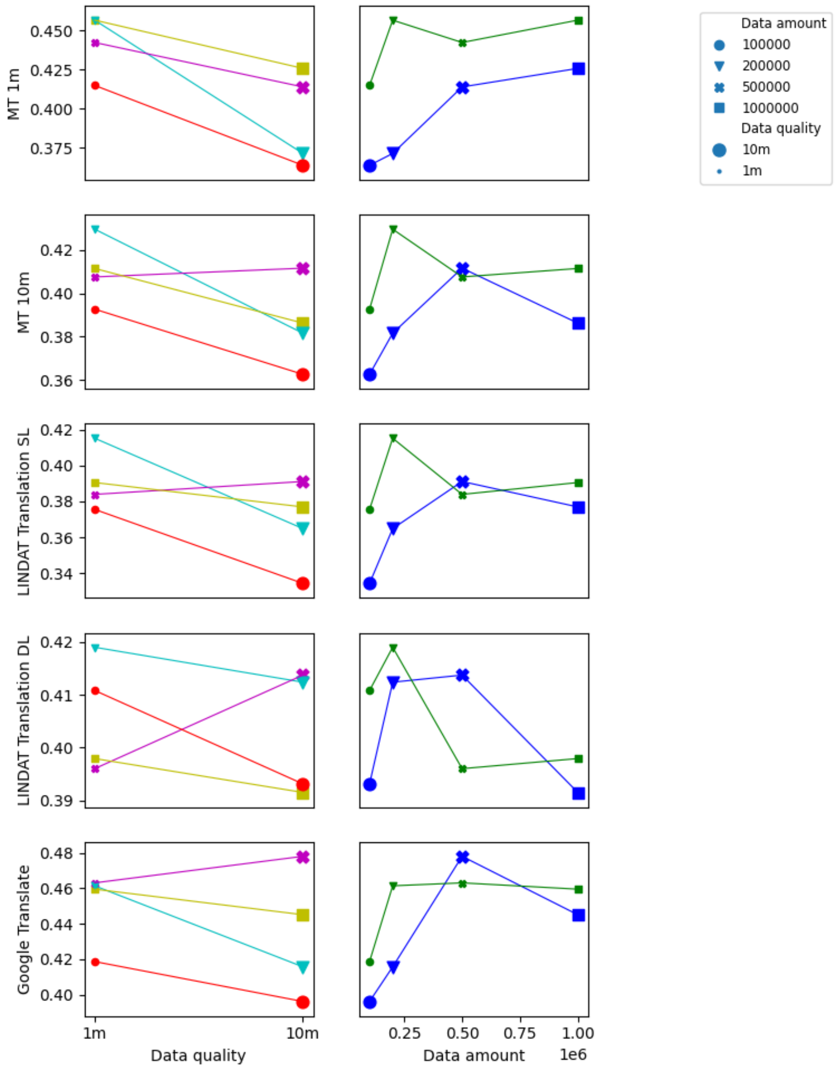

Figure 4.15: Pearson correlation between predictions generated by PredEst models with targets for test set Antrecorp from IWSLT translated by MT 1m, MT 10m, LINDAT Translation sentence-level model, LINDAT Translation document-level model and Google Translate

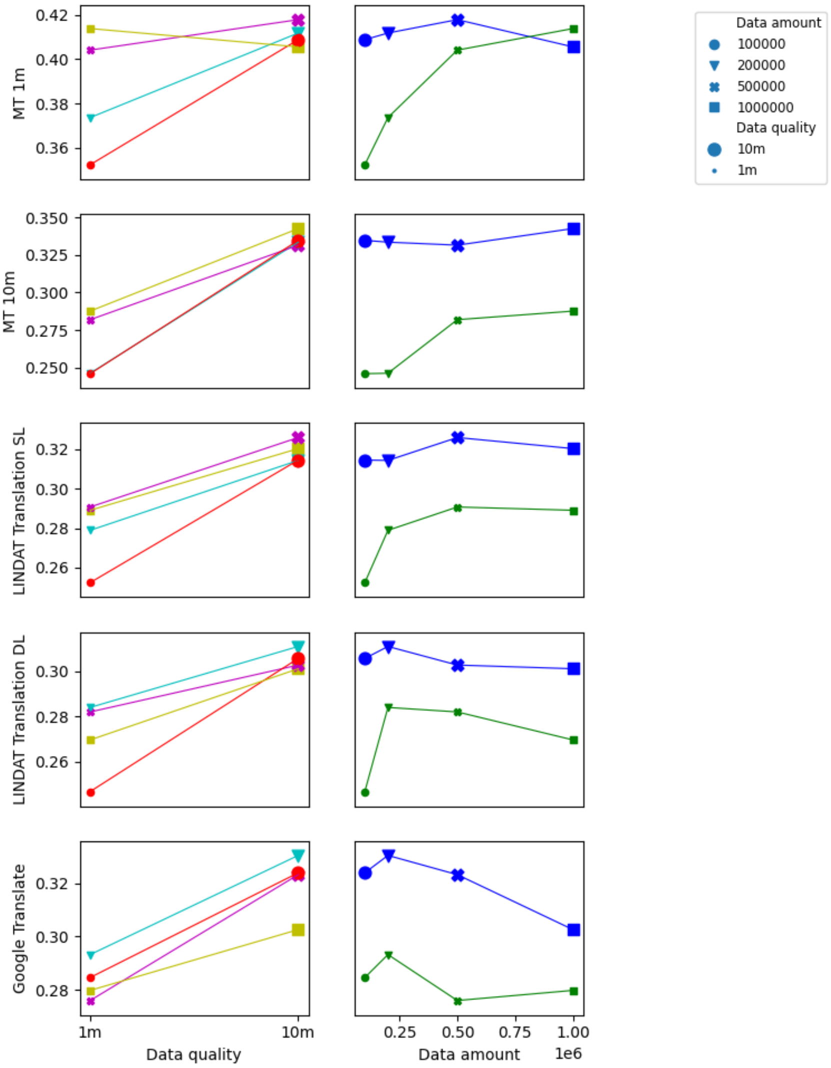

Figure 4.16: Pearson correlation between predictions generated by PredEst models with targets for test set Khan Academy from IWSLT translated by MT 1m, MT 10m, LINDAT Translation sentence-level model, LINDAT Translation document-level model and Google Translate

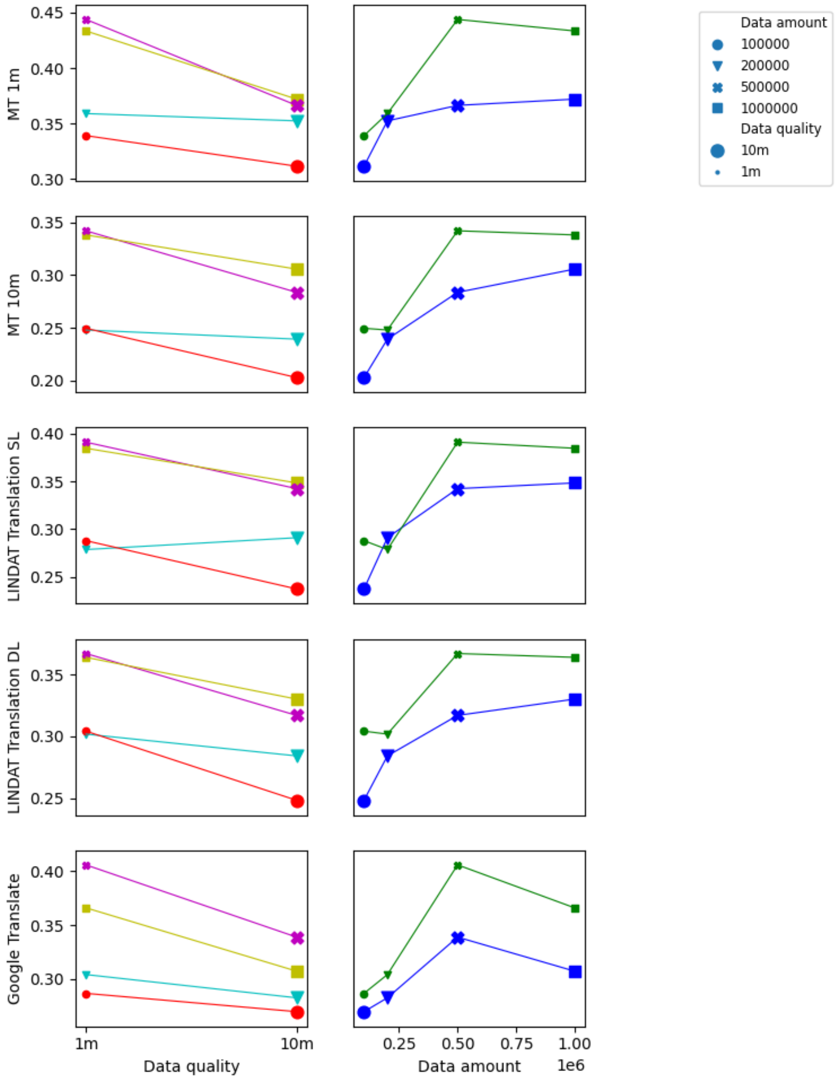

Figure 4.17: Pearson correlation between predictions generated by PredEst models with targets for test set SAO from IWSLT translated by MT 1m, MT 10m, LINDAT Translation sentence-level model, LINDAT Translation document-level model and Google Translate

Conclusion

The goal of the thesis was to study the relationship between MT and QE systems. We defined the term power of the MT system and explained how it is measured and what affects it. We surveyed related papers featuring the experiments that involve interaction between MT and QE of different power. We defined the terms power of the QE system and reviewed existing QE implementations.

We completed experiments that involved training multiple MT systems with varying power and multiple QE systems of varying power. We made the evaluation of QE systems on test sets including WMT18 News Translation Task test set translated by submitted MT systems and IWSLT test sets translated by our MT systems and online MT systems Google Translate and LINDAT Translation. We reviewed how different systems' dimensions that contribute to either MT or QE power affect evaluation results. From the experiments, we can conclude the following:

- QE systems perform better when trained on lower quality data, which holds for both high-power and low-power QE systems.
- QE systems performance is lower when evaluating high quality MT translations. This holds for both high-power and low-power QE systems.
- When evaluating high quality MT translation, the gap between low data quality QE systems and high data quality QE system performance is getting smaller.
- High-power QE systems work better for out-of-domain distribution than low-power QE systems, which fail to yield good quality estimation.

Bibliography

Sweta Agrawal, George Foster, Markus Freitag, and Colin Cherry. Assessing reference-free peer evaluation for machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1158–1171, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.91. URL `https://aclanthology.org/2021.naacl-main.91`.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.iwslt-1.1. URL `https://aclanthology.org/2020.iwslt-1.1`.

Yamini Bansal, B. Ghorbani, Ankush Garg, Biao Zhang, Maxim Krikun, Colin Cherry, Behnam Neyshabur, and Orhan Firat. Data scaling laws in nmt: The effect of noise and architecture. *ArXiv*, abs/2202.01994, 2022.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland, aug 23–aug 27 2004. COLING. URL `https://aclanthology.org/C04-1046`.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 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, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2301. URL `https://aclanthology.org/W16-2301`.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6401. URL `https://aclanthology.org/W18-6401`.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, Yujia Liu, and Shujian Huang. HW-TSC's participation at WMT 2021 quality estimation shared task.

In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.92`.

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.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3348. URL `https://aclanthology.org/W14-3348`.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzman, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555, 09 2020. doi: 10.1162/tacl_a_00330.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej 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*, pages 733–774, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.73`.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016.

B. Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier García, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. *ArXiv*, abs/2109.07740, 2021.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.478. URL `https://aclanthology.org/2021.emnlp-main.478`.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23:3 – 30, 2015.

Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone, and Rico Sennrich. The University of Edinburgh's submissions to the WMT18 news translation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 399–409, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6412. URL `https://aclanthology.org/W18-6412`.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P18-4020`.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics–System Demonstrations*, pages 117–122, Florence, Italy, July 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P19-3020`.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17:1–22, 09 2017a. doi: 10.1145/3109480.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark, September 2017b. Association for Computational Linguistics. doi: 10.18653/v1/W17-4763. URL `https://aclanthology.org/W17-4763`.

Tom Kocmi, Martin Popel, and Ondrej Bojar. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*, 2020.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.57`.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL `https://aclanthology.org/N03-1017`.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In

Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://aclanthology.org/P07-2045`.

Chi-kiu Lo. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5358. URL `https://aclanthology.org/W19-5358`.

Dominik Macháček, Jonáš Kratochvíl, Tereza Vojtěchová, and Ondřej Bojar. A speech test set of practice business presentations with additional relevant texts. In *Statistical Language and Speech Processing: 7th International Conference, SLSP 2019, Ljubljana, Slovenia, October 14–16, 2019, Proceedings*, page 151–161, Berlin, Heidelberg, 2019. Springer-Verlag. ISBN 978-3-030-31371-5. doi: 10.1007/978-3-030-31372-2_13. URL `https://doi.org/10.1007/978-3-030-31372-2_13`.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.566. URL `https://aclanthology.org/2021.acl-long.566`.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. Unbabel's participation in the WMT16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 806–811, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2387. URL `https://aclanthology.org/W16-2387`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Martin Popel. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6424. URL `https://aclanthology.org/W18-6424`.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL `https://aclanthology.org/W15-3049`.

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`.

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*, pages 5070–5081, Barcelona, Spain (Online), December 2020a. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.445. URL `https://aclanthology.org/2020.coling-main.445`.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online, November 2020b. Association for Computational Linguistics. URL `https://aclanthology.org/2020.wmt-1.122`.

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`.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL `https://aclanthology.org/2020.acl-main.704`.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://aclanthology.org/P16-1162`.

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 2006a. Association for Machine Translation in the Americas. URL `https://aclanthology.org/2006.amta-papers.25`.

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 2006b. Association for Machine Translation in the Americas. URL `https://aclanthology.org/2006.amta-papers.25`.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain, May 14–15 2009. European Association for Machine Translation. URL https://aclanthology.org/2009.eamt-1.5.

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://www.aclweb.org/anthology/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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Brian Thompson and Matt Post. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.8. URL https://aclanthology.org/2020.emnlp-main.8.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. QEMind: Alibaba's submission to the WMT21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 948–954, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wmt-1.100.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL https://arxiv.org/abs/1609.08144.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. IST-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.102`.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

List of Figures

List of Tables