FRI Natural language processing course 2021
Project report

University *of Ljubljana*
Faculty *of Computer and*
Information Science

Neural machine translation

Eva Boneš, Teja Hadalin, Meta Jazbinšek, Sara Sever, Erika Stanković

Abstract

In this paper, we present our work on a neural translation model, specialized in translating English TED Talks into Slovene. The aim is to improve the transcription of the speeches to reach a wider audience with the option of automatic subtitling. First, we trained a transformer model on the general data, a collection of corpora from the Opus site, and then fine-tuned it on a specific domain which was a corpus of TED Talks. To see the functionality of the model, we carried out the evaluation of the pretrained, general and domain version of the model. We evaluated the translations with automatic metrics and manual methods – the adequacy/fluency and the user feedback criterion. Analysis of the results shows that the translation model did not produce expected results and it could not be used to translate speeches in real life. However, the translations did convey the main message of the texts. In the future attempts, we should consider improvements, such as including more data which covers only one specific topic.

Keywords
neural machine translation, English to Slovene, transformer, fairseq, evaluation, adequacy, fluency, user feedback

Advisors: Slavko Žitnik, Špela Vintar, Mojca Brglez

1. Introduction

Machine translation (MT) is the problem of automatically translating text from one language to another. It has been researched since the 1950s, but only recently, with the rise of deep learning, did it prove to be solvable, although the possibility of achieving fully automatic machine translations of high quality is still being questioned.

In this paper, we trained a transformer model from scratch on a large general corpus, which we then fine-tuned on a corpus consisting of TED Talks to make a model specialized for the translation of transcribed speeches. We also found a pretrained model for the baseline to which we were able to compare our translation models. We automatically and manually evaluated all three models on the validation datasets constructed from TED Talks, as well as on the validation dataset provided by our assistant. We also evaluated the general translation model on the validation dataset constructed from the large general corpus.

In Section 2, we first describe the data we used. In the subsequent Section, we describe all the methods for both training and evaluating the models. Later on, in Sections 4 and 5, we present the results and discuss them.

1.1 Related works

There are three main approaches to solving the problem, all with their advantages and shortcomings. Rule-based machine translation (RBMT) is the oldest of the bunch and it requires expert knowledge about both the source and the target language to develop syntactic, semantic and morphological rules. In contrast to other two approaches, RBMT does not need bilingual data and when a set of rules is set for a particular language it can be reused for pairings with different languages. Its main disadvantage is that it requires a lot of expertise to manually set rules and find good dictionaries which can take a lot of time or might even border on impossible in some languages.

Another approach, which gained popularity in the 1990s, uses statistical models based on the analysis of bilingual text corpora. The idea behind statistical machine translation (SMT) as proposed in [1] is, if given a sentence T in the target language, we seek the sentence S from which the translator produced T – we want to choose S so that it maximizes $Pr(S|T)$. Comparing to RBMT, this approach needs less manual work from linguistic experts and we can achieve more fluent translations (provided we have a good language model). STM models are also generally built in a way that one can be used on more language pairs, but this requires a bilingual corpus which can be hard to find, depending on the language pair we are interested in. This model is also less suitable for language pairs with big differences in word order.

As with many computer science fields, the current state-of-the-art approaches are based on neural networks. Contrary to building a pipeline of separate tasks with RBMT and STM, we only need to build one network with neural machine translation (NMT). The biggest challenge in building a successful English to Slovene (or vice-versa) automatic translator is obtaining a sufficiently big bilingual corpus. Like all deep learning approaches, having a large and quality dataset is crucial for the success of the model. To deal with this exact problem, a lot of approaches of pre-training a network on monolingual data (that can be obtained easily) have been proposed.

Bidirectional Encoder Representations from Transformers (BERT) [2] which was proposed by researchers at Google, uses two strategies to deal with the problem, namely masked language modeling (MLM) and next sentence prediction (NSP). With MLM before feeding sequences into the model, 15% of the words are replaced with a [MASK] token, and the model then attempts to reproduce the original value of sequences. In addition to a bigger dataset we can get this way, the model achieves more context awareness and when fine-tuned produces more fluent translations. In NSP, the model receives pairs of sentences as input and learns to predict if the second sentence is a subsequent sentence of the first one. Similar to MLM, the model achieves more context-awareness.

In 2020, mRASP [3], that could be very useful in our case, has been proposed. The authors built a pretrained NMT model that could be fine-tuned for any language pair. They used 197M pairs of sentences, which is much more than we could obtain for only English-Slovene translations.

Although these methods have proven to be successful, one of today's largest available databases of pretrained translation models was trained using just a standard transformer model and still achieved great results. The Tatoeba Translation Challenge [4] aims to provide data and tools for creating state-of-the-art translation models. The focus is on low-resource languages to push their coverage and translation quality. It currently includes data for 2,963 language pairs covering 555 languages. Along with the data, they also released pretrained translation models for multiple languages which are regularly updated.

1.2 Frameworks

fairseq [5]

fairseq is a sequence modeling toolkit written in PyTorch for training models for translation, summarization, and other natural language processing tasks. It provides different neural network architectures, namely convolutional neural networks (CNN), Long-Short-Term Memory (LSTM) networks, and Transformer (self-attention) networks. The architectures can be configured to our own needs and many implementations for different tasks have been proposed since the fairseq's introduction in 2019. In addition to different architectures, they also provide pretrained models and preprocessed test sets for different tasks, but none of them is in Slovene, so we will not be able to use them.

Marian NMT [6]

Marian is an efficient Neural Machine Translation framework written in pure C++. It is developed by Microsoft and it provides fast multi-GPU training and batched translation on GPU/CPU. It allows training on raw texts using built-in SentencePiece [7] (an unsupervised text tokenizer and detokenizer). It offers multiple different models: deep RNNs with Deep Transition Cells [8], transformer models [9], multi-source models [10], RNN, and transformer-based language models. The code is publicly available on https://marian-nmt.github.io/.

T5 [11]

T5 (Text-To-Text Transfer Transformer) is a framework, developed by Google, that trains a single model for a variety of text tasks (translation, summarization...). Its performance is comparable to task-specific architectures and at that time it was considered state-of-the-art when combined with scale. It uses an encoder-decoder model and it was developed using Tensorflow. For tokenizing, it also uses the SentencePiece model [7]. The code is publicly available on https://github.com/google-research/text-to-text-transfer-transformer. The framework includes some pretrained models, but because of the vocabulary they used, it can only process a predetermined, fixed set of languages. This means that for our task of translating into Slovene we would need to train the model from scratch. The framework needs a significant amount of preprocessing because it requires a specific format of input data: <input>\t<target>.

2. Data

2.1 General translation model

The datasets for the general translation model are the eight biggest corpora from the Opus site (https://opus.nlpl.eu [12]) for the Slovene-English language pair. The exact size of each one, complete with the number of tokens, links, sentence pairs, and words, is noted in Table 1. The corpora were chosen based on the quantity of the data, so the general translation model would contain a large amount of diverse information. After a brief look at the contents of each one, we can see that some datasets are of higher quality and more reliable, because of the source of the original texts and their translations, for example the corpora from European institutions.

Europarl is a parallel corpus from the European Parliament website by Philipp Koehn (University of Edinburgh) and is extracted from the proceedings of the European Parliament, from 1996–2011. The **DGT** corpus is a collection of translation memories from the European Commission's Directorate-General for Translation. The aligned translation units in the DGT have been extracted from one of its large shared translation memories in EURAMIS (European advanced multilingual information system).

CORPUS	Tokens	Links	Sentence pairs (MOSES format)	Words (MOSES format)
Europarl.en-sl	31.5 M	0.6 M	624,803	27.56 M
CCAligned.en-sl	131.3M	4.4 M	4,366,555	110.08 M
DGT.en-sl	215.8M	5.2 M	5,125,455	162.58 M
MultiCCAligned.en-sl	5.6 G	4.4 M	4,366,542	110.01 M
OpenSubtitles.en-sl	178.0 M	2.0 M	19,641,477	213.00 M
TildeMODEL.en-sl	2305.4 M	21.1 M	2,048,216	79.90 M
WikiMatrix.en-sl	1.1 G	0.9 M	318,028	11.99 M
wikimedia.en-sl	350.6 M	31.8 K	31,756	1.50 M
XLEnt.en-sl	200.7 M	0.9 M	861,509	4.53 M

Table 1. Size of datasets for the general translation model.

The other corpora are a collection of translations from different Internet sources, which makes them less reliable, however, they are still very valuable because they ensure a large quantity of the data.

The **CCAligned** corpus consists of parallel or comparable web-document pairs in 137 languages aligned with English. These web-document pairs were constructed by performing language identification on raw web documents, and ensuring corresponding language codes were corresponding in the URLs of web documents. The **MultiCCAligned v1** corpus has been further processed for making it a multi-parallel corpus by pivoting via English. The **OpenSubtitles** is a corpus compiled from a large database of movie and TV subtitles from the http://www.opensubtitles.org/ repository. Based on their website, there are 745,363 subtitles for English and 72,819 for the Slovene language currently uploaded. From a linguistic perspective, subtitles cover a wide and interesting width of genres, from colloquial language or slang to narrative and expository discourse (as in e.g. documentaries) [13]. In **Tilde MODEL** corpus there are over 10M segments of multilingual open data for publication on the META-SHARE repository, maintained by the Multilingual Europe Technology Alliance, and on the EU Open Data Portal. The data has been collected from sites allowing free use and reuse of its content, as well as from Public Sector web sites. **WikiMatrix v1** is a parallel corpus from Wikimedia compiled by Facebook Research. The approach is based on multilingual sentence embeddings to automatically extract parallel sentences from the content of Wikipedia articles in 85 languages. To get an indication of the quality of the extracted bitexts, neural MT baseline systems are trained on the mined data only for 1,886 language pairs, and evaluated on the TED corpus, achieving strong BLEU scores for many language pairs [14]. The **Wikimedia v20210402** corpus contains Wikipedia translations published by the Wikimedia Foundation and their article translation system. The **XLEnt v1** corpus was created by mining CCAligned, CCMatrix, and WikiMatrix parallel sentences. These three sources were extracted from the web data from Commoncrawl Snapshots and Wikipedia snapshots. Entity pairs were obtained by performing named entity recognition and typing on English sentences and projecting labels to non-English aligned sentence pairs. The lexical/semantic/phonetic alignment approach yielded more than 160 million aligned entity pairs in 120 languages paired with English.

2.2 Domain translation model

Our domain translation model is specialized in translating TED Talks. The main goal of this project is to provide a useful and effective tool for translating speech, especially TED Talks, and this way ensuring access to a wide range of talks and other speeches in the Slovene language. Despite being aware of the countless enhancements necessary to provide an excellent end product, our focus, in this fairly short time frame, is to provide a machine translator that will work effectively and offer comprehensible and useful translations.

For the domain-specific machine training, we opted for the two TED Talk corpora accessible on the Opus website – the TED2013 and TED2020 corpus. The included texts are mainly transcripts of speeches on various topics and their Slovene translations. Both datasets add up to 1.8 million words (MOSES format) and 2.1 million tokens which is enough to form a well-rounded base for machine learning. For more information about the domain-specific corpora see Table 2.

We expanded the datasets by manually aligning 15 TED Talks from 2018 and 2019 that are available on the TED website (https://www.ted.com/talks).

CORPUS	Tokens	Links	Sentence pairs (MOSES format)	Words (MOSES format)
TED2013	0.5 M	15.2 k	14,960	0.45 M
TED2020	1.6M	43.9 k	44,340	1.35 M
Extras	23005	/	983	/

Table 2. Size of datasets for the domain translation model.

3. Methods

3.1 Pretrained model

As a baseline for evaluating our models, we found an already trained model, available in HuggingFace (https://huggingface.

co/Helsinki-NLP/opus-mt-en-zls) [4]. It is a transformer-based multilingual model that includes all the South Slavic languages. The framework provides both the South-Slavic to English model and the English to South-Slavic model. On the Tatoeba test dataset for Slovene, the English to South-Slavic (en-zls) model achieves an 18.0 BLEU score and 0.350 chr-F score.

The model in question was trained using MarianNMT [6]. The authors applied a common setup with 6 self-attentive layers in both, the encoder and decoder network using 8 attention heads in each layer. SentencePiece [7] was used for the segmentation into subword units.

The translation model can be loaded through the *transformers* library in Python and for translation into Slovene, we must add the Slovene language label at the beginning of each sentence (>>slv<<).

3.2 Training from scratch

For training our model from scratch, we have decided to use an extension of fairseq [15] that has additional data augmentation methods. We have trained our general model on a corpus described in Subsection 2.1.

3.2.1 Preprocessing

Before training the model, we had to preprocess the data. The datasets were already formatted as raw text with one sentence per line and with lines aligned in English and Slovene datasets. We first normalized the punctuations, removed non-printing characters and tokenized both corpora with Moses tokenizer [16]. We removed all the sentences that were too short (2 tokens or less) or too long (250 tokens or more) and the ones where the ratio of lengths was larger than 5 because there is a good chance that this kind of sentences are not translated properly. We then applied Byte pair encoding (BPE) [17] to the dataset. The algorithm learns the most frequent subwords to compress the data and thus induces some tokens that can help recognize less frequent and unknown words.

With this preprocessed data, we then built the vocabularies that we used for training and binarized the training data. Cleaned and preprocessed training data has $\approx 16M$ sentences with $\approx 345M$ tokens in English and $\approx 341M$ in Slovene. Both of the vocabularies have around 45000 types.

3.2.2 Training

We trained a transformer [18] model with 5 encoder and 5 decoder layers in the fairseq framework. We used Adam optimizer, inverse square root learning rate scheduler with an initial learning rate of $7e^{-4}$ and dropout. We also used the proposed augmentation with cut-off augmentation schema that randomly masks words and this way produces more training data and a more robust translator. The exact configuration is shown in Appendix A.

We trained our model for 8 epochs with the before-mentioned initial learning rate, after which the minimum loss scale (0.0001) was reached, meaning that our loss was probably exploding. We tried training one more epoch with a lower initial learning rate, but obtained a worse performance, and then the minimum loss scale was reached again. Results of all the epochs are shown in Chapter 4.

3.3 Fine-tuning on TED talks

We preprocessed the TED data in the same way as the general, only this time we used the same dictionary as before and we did not build a new one. Less than 0.1% of tokens in training and validation sets were replaced with *unknown* tokens, so our original dictionary was evidently large enough. We used the best performing epoch from our general translation model (according to the loss on our validation set) for fine-tuning it on our domain data. We trained three different models with three slightly different configurations – one with the same augmentation parameters as the general model, one with increased masking probability and decreased dropout and initial learning rate, and one without augmentation. The exact configurations are shown in Appendix B. We trained all of the models for 100 epochs and we are presenting the results of the best epoch for each of them.

3.4 Evaluation

In order to test the performance of the general translation model and the fine-tuned translation model for TED Talks, we had to evaluate the translations.

The automatic evaluation was carried out on three validation sets. First, the general translation model was evaluated on a subset of the general training data (hereinafter referred to as the general validation set). All three models were evaluated on a subset of the domain training data (hereinafter referred to as the domain validation set), as well as the validation set provided by our assistant (hereinafter referred to as the assistant validation set). Manual evaluation was only performed on a subset of the domain validation set, as described in 3.4.2.

3.4.1 Automatic evaluation

Since the manual evaluation of the translations is very time-consuming, it is very difficult to evaluate a sufficient amount of sentences this way. In cases like this, automatic evaluation metrics are often used. Because natural language is quite subjective, the perfect measure does not exist, but by evaluating our results with different techniques, we will be able to assess how successful our translation model is and to compare it with others. In the following subsections, we describe some of the more important metrics that were used. The metrics were calculated using the *nltk* [19] and *jiwer* libraries in Python.

Bilingual evaluation understudy (**BLEU**) [20] score is the most commonly used score for comparing different approaches for automatic translation. It tests the quality of machine-translated texts by comparing them to a professional human translation. The algorithm operates by counting matching n-grams between the candidate and reference translations where matches are positionally independent. The higher the number of matches, the higher the BLEU score.

Metric for Evaluation of Translation with Explicit ORdering (**METEOR**) [21] was designed to fix some problems found in BLEU. It is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. Compared to BLEU, it achieved a higher correlation with human judgment.

NIST metric is based on BLEU, but with some differences. For example, when calculating n-gram precisions, NIST also calculates how informative a particular n-gram is, meaning that rarer correctly matched n-grams are weighted more, whereas in BLUE, all are weighted equally.

Character n-gram F-score (**chrF**) [22] is based on character n-gram precision and recall enhanced with word n-grams. The tool calculates the F-score averaged on all character and word n-grams, where the default character n-gram order is 6 and word n-gram order is 2. The arithmetic mean is used for n-gram averaging.

GLEU [23] is an SVM-based machine learner based on a metric that estimates fluency by examining the use of parser outputs as metrics. The authors show that the machine learner provides a consistent estimator of fluency.

Word error rate (**WER**) is a metric derived from the Levenshtein distance. The method compares two sequences on word level and the value represents the ratio of substituted, deleted and inserted words to correct (same) words. In contrast to other metrics, lower WER is better than higher.

3.4.2 Manual evaluation

The texts were also evaluated manually, namely by the fluency-adequacy criterion. For this part of the evaluation, the Excel format was used. We chose 6 paragraphs containing 10 consecutive segments to ensure that the context was clear. Each of the three evaluators (the translators from our group) were designated 20 segments. To determine the adequacy of the translation, the evaluator marks how much of the meaning expressed in the source text is also expressed in the target translation. To determine the fluency of the translation, the evaluator marks whether the translation is grammatically well-formed, whether it contains correctly spelled words, whether it is intuitively acceptable and can be sensibly interpreted by a native speaker. To test the adequacy, the evaluator compares the source text and the translation and, to check the fluency, they focus merely on the translation. The evaluators gave the answers to both questions on a scale from 1 to 4. We chose this evaluation technique because it describes the quality of the translation clearly and simply. Because we evaluated three different translation models (pretrained, general and domain), we needed to evaluate the same segments of texts three times. Evaluating one text multiple times by the same person is not recommended, therefore, we switched the translations between the members when starting the evaluation of each translation model.

3.4.3 End user comprehensibility questionnaire

Finally, we evaluated the domain machine-translated texts from the end-user's point of view. Evaluators who were not familiar with this project were given the translated texts from the domain model and a questionnaire formed by the translation team of this project. The objective of the questionnaire was whether the end-users understand the information given in the text, meaning it tested the translation's functionality. The questionnaire was given to nine persons and each one had to evaluate 20 segments, so in the end, we obtained three evaluations for each text. It included the following questions:

1. How comprehensible is the text?
2. To what degree does the text seem like it was produced by a native speaker of Slovene?
3. How would you grade the text as a whole?
4. What is the main message of the text?
5. What do you see as the most problematic part of the text?

For the first and the second question, the end-users answered on a scale from 1 to 4, 1 meaning 'not at all' and 4 meaning 'very much'. The third answer had to be a number from 1 to 4. For the fourth question, they had to answer with one sentence, and for the fifth question, they had to choose between the following answers: 'unknown words', 'too little context', 'wrong syntax' and 'other'. We chose this evaluation technique because it shows whether the translation is in fact functional and useful. The evaluation results were interpreted by the translators from our team.

4. Results

For the training of our models, we used the Slovenian national supercomputing network that provides access to cluster-based computing capacities. We used the Arnes cluster that is equipped with 48 NVIDIA Tesla V100S PCIe 32GB graphic cards. When training on two of them with batch size of 8000 tokens, one epoch took approximately 4 hours for the general translation model and one minute for fine-tuning it on TED data.

4.1 Examples of translation

Input English sentence: *Obviously, these guys are about slow, careful analysis, and that's because the game has outgrown the days where pure street smarts and people-reading can get you to the top.*
Reference Slovene translation: *Očitno je, da imajo ti fantje raje počasno, previdno analizo, to pa zato, ker je igra prerasla dni, ko si lahko z ulično pretkanostjo in branjem ljudi, prišel do vrha.*
Slovene translation using the best-performing general translation model: *Očitno, ti fantje so o počasni, skrbni analizi, in to je zato, ker je igra prerasla dni, ko čiste ulične pametnice in ljudje - branje lahko dobite na vrh.*
Slovene translation using the best-performing domain translation model: *Očitno so ti fantje počasni, previdni analizi in to je zato, ker je igra zaraščala dni, ko vas lahko čisti ulični strmeči in branje ljudi pripelje do vrha.*

Dataset	Metric	Pretrained	General (epochs)								Domain		
			1	2	3	4	5	6	7	8	Configuration 1	Configuration 2	Configuration 3
Assistant	BLEU	0.335	0.342	0.358	0.361	0.364	0.364	0.370	0.371	**0.373**	0.338	0.345	0.276
	chr-F	**0.592**	0.564	0.574	0.579	0.577	0.581	0.583	0.585	***0.586***	0.562	0.564	0.514
	GLEU	0.350	0.352	0.366	0.370	0.372	0.373	0.376	0.378	**0.379**	0.350	0.356	0.293
	METEOR	**0.509**	0.483	0.496	0.501	0.498	0.502	0.506	0.506	***0.508***	0.480	0.485	0.427
	NIST	7.798	7.655	7.867	7.960	7.945	7.979	8.005	8.004	**8.059**	7.656	7.724	6.67
	WER	0.548	0.560	0.546	0.539	0.539	0.537	0.538	0.536	**0.534**	0.556	0.552	0.617
General	BLEU	-	0.387	0.398	0.405	0.409	0.411	0.417	0.417	**0.420**	-	-	-
	chr-F	-	0.606	0.616	0.619	0.624	0.625	0.629	0.629	**0.629**	-	-	-
	GLEU	-	0.391	0.401	0.407	0.411	0.413	0.417	0.417	**0.420**	-	-	-
	METEOR	-	0.545	0.556	0.560	0.565	0.566	0.569	0.569	**0.571**	-	-	-
	NIST	-	8.752	8.922	8.987	9.063	9.096	9.144	9.114	**9.177**	-	-	-
	WER	-	0.518	0.508	0.503	0.501	0.496	0.497	0.498	**0.494**	-	-	-
Domain	BLEU	**0.192**	0.155	0.167	0.168	0.171	0.175	0.175	0.168	0.179	***0.182***	0.173	0.114
	chr-F	**0.514**	0.487	0.496	0.495	0.497	0.500	0.498	0.500	***0.505***	0.503	0.497	0.440
	GLEU	**0.230**	0.201	0.211	0.212	0.214	0.217	0.218	0.213	0.222	***0.224***	0.216	0.167
	METEOR	0.420	0.398	0.407	0.409	0.409	0.414	0.412	0.416	0.420	**0.426**	0.416	0.346
	NIST	**5.481**	4.877	5.067	5.105	5.132	5.151	5.179	5.074	5.230	***5.344***	5.209	4.228
	WER	**0.659**	0.711	0.696	0.694	0.690	0.689	0.689	0.698	0.685	***0.667***	0.680	0.756

Table 3. Evaluation scores for all models and all validation datasets.

4.2 Automatic evaluation results

In Table 3, we present the quantitative results of the automatic evaluation for the pretrained, general and domain models.

4.3 Manual evaluation results

Along with the automatic evaluation metrics, we also performed manual evaluation which, despite being carried out on a small number of data, provided a valuable human insight into the final product and a better understanding of the typology of the mistakes that occurred in the translations.

Each validation set was assessed by two evaluators at all three stages of the model development. The results presented in Table 4 represent the average value of the fluency and adequacy rates for the pretrained, general and domain model, respectively.

MODEL	Fluency	Adequacy
Pretrained	2.99	3.09
General	2.83	2.9
Domain	2.71	2.9

Table 4. Manual evaluation results on TED validation set.

4.4 End-user comprehensibility questionnaire results

Additionally, we received feedback from the end-users based on the questionnaire for the texts from the domain translation model. The average score of the answers which could be interpreted numerically (questions 1, 2 and 3 already stated in subsection 3.4.3) is presented in Table 5. Based on the answers to the question 'What is the text's main message?', the users have, for the most part, understood the text to a degree that they could sufficiently summarize the point of it. The most frequent answer to the last question (What is the most problematic part of the text?) was wrong syntax, followed by the lack of context and then the unknown words, and they also pointed out the general structure of the text was confusing.

Text	Question 1	Question 2	Question 3
1.	1.33	1	1
2.	2	1.33	1.33
3.	3	2	2.33
4.	1.66	1	1.33
5.	2	1.66	1.66
6.	2.33	1.66	2
All:	2.053333333	1.441666667	1.608333

Table 5. End-user feedback.

5. Discussion

Looking at the results in Table 3, we can first see that on assistant and general validation sets the final epoch of our general model performs the best according to most metrics. This is expected, as the assistant validation set is comprised of the data from the corpora that we used for training, so our model may be overfitted on this dataset. The same goes for the general dataset that comes from the same corpora and even though the sentences are not the same, the style is.

Connected to this, all of the results in the domain validation set are considerably worse than in other two datasets. We can account this to the fact that the domain validation set is the only one that is truly different from the main training data. As to why the pretrained model in most aspects performs better than our fine-tuned model, we assume this is because our domain data is not specific enough. We, therefore, could not really fine-tune our model to any specific styles or words, nor were we able to do that in the validation set. The pretrained model performs better because it is trained on more data than our domain model is fine-tuned on – the TED corpus is relatively small even though we included some additional texts.

Similarly, the results of the manual evaluation showed that the pretrained model produced the most fluent translations with the average score of 2.99 out of 4. This model also achieved the highest score in the adequacy criterion. If we take a closer look at the results of the other two models, it can be seen that both models faced similar difficulties in translating phrasal verbs, terminology, word order and other lexical structures. The manual evaluation results are relatively low: the general and the domain model received the average of less than 3 points, in both fluency and adequacy.

However, a quick analysis of the evaluation rates showed that the lowest ratings appeared in segments with specialised vocabulary, for example the text describing fingerprints in detail or a deadly accident, and in segments with a higher register, for example the eloquent text on immigrants. In both cases, the rate was never lower than 2.8. The highest rated segments (with the score above 3) included short and simple sentences with everyday vocabulary. Based on the evaluation results, it seems that our domain model would be more valuable in translating general texts with neutral style and vocabulary. The group members that evaluated these segments had been participating in this project from the very beginning, so it was crucial to obtain a more objective assessment of our models.

Looking at the results from Table 5, the gathered feedback from the questionnaire revealed that overall, the end-users thought that the texts are comprehensible to a certain degree, but do not at all seem like they were produced by a native speaker of Slovene. We can see that the highest average grade for a specific text is 2.33, while the lowest is 1. This variation occurs because not all of the chosen texts were equally complex. The fact that the end-users were in almost all cases able to clearly summarize the main message in one sentence, shows that comprehension of the text was still possible despite of a large number of significant mistakes (wrong syntax, unknown words, lack of context, changing genders etc.).

The goal of the project was to produce a machine translator that would automatically create Slovene subtitles for English TED Talks, so that a wider audience could understand the

the topics addressed in the speeches. Unfortunately, the final version did not meet our expectations regarding the quality of the translations. Some of the major flaws that appeared in the translations were wrong syntax, untranslated words, incomprehensible structures, wrong use of terminology and wrong translations of polysemes. While we expected the translator to be insufficient for translating complex sentences, we were surprised that it did not do well when translating even fairly basic structures.

The reasons for poor machine translations could be numerous. We might have not collected enough data or the chosen data might not have been the most suitable. We estimate that the main factor that could have impacted the results is the wide range of different topics covered in TED Talks. Our domain translator did not focus on one domain and essentially there was not enough specific data from which it could train. What is more, the initial data consisted of transcriptions of English spoken discourse and Slovene translations in the form of subtitles. Neither spoken discourse nor subtitles have characteristics typical for standard text types. Finally, not all of the chosen texts were equally complex and had different syntactic and lexical features. Therefore, some of the texts in the data were too difficult to translate.

6. Conclusion

The main purpose of this project was to develop a tool that could automatically provide Slovene subtitles for English TED Talks. Our final model conveys the main message of the texts and is based on appropriate methodology. The results of automatic metrics showed that it is comparable to other neural machine translation models. On the other hand, the lack of uniform training dataset resulted in poor translations. However, we believe that only a few improvements could help us reach better results. Neural machine translation is still relatively new and will develop in the following years because it is useful for translators and general public. Our project contributed to the advancement of the field and could provide valuable information for similar work in the future.

Acknowledgments

We would like to thank our mentors, Slavko Žitnik, Špela Vintar and Mojca Brglez, for helping us with the project. We would also like to thank the nine evaluators who provided end-user feedback by filling out our questionnaire.

We would also like to thank SLING for giving us access to powerful graphic cards to successfully finish our training, as we would still be training our general model without them. Special thanks to Barbara Krašovec from Arnes support who helped us with our numerous problems when trying to connect to their cluster.

References

[1] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, “A statistical approach to machine translation,” *Comput. Linguist.*, vol. 16, no. 2, p. 79–85, Jun. 1990.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[3] Z. Lin, X. Pan, M. Wang, X. Qiu, J. Feng, H. Zhou, and L. Li, “Pre-training multilingual neural machine translation by leveraging alignment information,” 2021.

[4] J. Tiedemann, “The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT,” in *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1174–1182. [Online]. Available: https://www.aclweb.org/anthology/2020.wmt-1.139

[5] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[6] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, “Marian: Fast neural machine translation in C++,” in *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia, 2018. [Online]. Available: https://arxiv.org/abs/1804.00344

[7] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” *CoRR*, vol. abs/1808.06226, 2018. [Online]. Available: http://arxiv.org/abs/1808.06226

[8] A. V. Miceli Barone, J. Helcl, R. Sennrich, B. Haddow, and A. Birch, “Deep architectures for neural machine translation,” in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 99–107. [Online]. Available: https://www.aclweb.org/anthology/W17-4710

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 06 2017.

[10] M. Junczys-Dowmunt and R. Grundkiewicz, “An exploration of neural sequence-to-sequence architectures for automatic post-editing,” 06 2017.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text

transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[12] J. Tiedemann, "Parallel data, tools and interfaces in opus," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

[13] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 923–929. [Online]. Available: https://www.aclweb.org/anthology/L16-1147

[14] H. Schwenk, V. Chaudhary, S. Sun, H. Gong, and F. Guzmán, "WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia," 2019.

[15] stevezheng23, "fairseq_extension." [Online]. Available: https://github.com/stevezheng23/fairseq_extension

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. 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*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 177–180. [Online]. Available: https://www.aclweb.org/anthology/P07-2045

[17] R. Sennrich, B. Haddow, and A. 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)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[19] E. L. Steven Bird and E. Klein, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[20] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," 10 2002.

[21] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72. [Online]. Available: https://www.aclweb.org/anthology/W05-0909

[22] M. Popović, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 392–395. [Online]. Available: https://www.aclweb.org/anthology/W15-3049

[23] A. Mutton, M. Dras, S. Wan, and R. Dale, "GLEU: Automatic evaluation of sentence-level fluency," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 344–351. [Online]. Available: https://www.aclweb.org/anthology/P07-1044

Appendix

A General model configuration

```
--arch transformer
--optimizer adam
--adam-betas ’(0.9, 0.98)’
--adam-eps 1e-9
--clip-norm 0.0
--augmentation
--augmentation_schema cut_off
--augmentation_masking_schema word
--augmentation_masking_probability 0.05
--augmentation_replacing_schema mask
--criterion
    label_smoothed_cross_entropy_with_regularization

--weight-decay 0.0001
--label-smoothing 0.1
--dropout 0.3
--attention-dropout 0.1
--activation-dropout 0.1
--lr-scheduler inverse_sqrt
--lr 7e-4
```

B Fine-tuning configurations

B.1 Configuration 1

```
--arch transformer
--optimizer adam
--adam-betas ’(0.9, 0.98)’
--adam-eps 1e-9
--clip-norm 0.0
--augmentation
--augmentation_schema cut_off
--augmentation_masking_schema word
--augmentation_masking_probability 0.05
--augmentation_replacing_schema mask
--criterion
    label_smoothed_cross_entropy_with_regularization

--weight-decay 0.0001
--label-smoothing 0.1
--dropout 0.3
--attention-dropout 0.1
--activation-dropout 0.1
--lr-scheduler inverse_sqrt
--lr 1e-6
```

B.2 Configuration 2

```
--arch transformer
--optimizer adam
--adam-betas ’(0.9, 0.98)’
--adam-eps 1e-9
--clip-norm 0.0
--criterion
    label_smoothed_cross_entropy_with_regularization

--weight-decay 0.0001
--label-smoothing 0.1
--dropout 0.3
--attention-dropout 0.1
--activation-dropout 0.1
--lr-scheduler inverse_sqrt
--lr 1e-6
```

B.3 Configuration 3

```
--arch transformer
--optimizer adam
--adam-betas ’(0.9, 0.98)’
--adam-eps 1e-9
--clip-norm 0.0
--augmentation
--augmentation_schema cut_off
--augmentation_masking_schema word
--augmentation_masking_probability 0.1
--augmentation_replacing_schema mask
--criterion
    label_smoothed_cross_entropy_with_regularization

--weight-decay 0.0001
--label-smoothing 0.1
--dropout 0.1
--attention-dropout 0.1
--activation-dropout 0.1
--lr-scheduler inverse_sqrt
--lr 1e-8
```