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

Automatic language translation

Miha Petrišič, Veno Gaube, Armin Komić, Jasna Cindrič, Rebeka Zajc, Gaja Černe, Eva Bolta, Vesna Železnik

Introduction

For our second assignment we outlined some of the key steps in the process of machine translation. First we will discuss our thought process behind choosing the right framework for building the translation model. We will describe the work that has already been done, that includes corpora selection and successfully building our first translation model on the subsection of our data. We will also provide some of the results of the current model and comments on how to improve our translations. In the end we will discuss some of the methods that we are considering for evaluating our results.

Different Methods

We learnt about five different modern methods that are currently being used in the field of automatic language translation. At the lectures we mentioned BERT and after doing a bit of research into it we got to the XLM-RoBERTa model [1]. The Facebook AI team released XLM-RoBERTa in November 2019, its sole training objective is the Masked Language Model. XLM-RoBERTa is a multilingual model trained on 100 different languages. Unlike some XLM multilingual models, it does not require lang tensors to understand which language is used, and should be able to determine the correct language from the input ids. It builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. After that we went through a few papers talking about NLP and Neural networks when we discovered the Marian NMT [2], which is an efficient free Neural Machine Translation framework written in C++. It is being developed by Microsoft and deployed by many companies, organizations and research projects. Notably it enables fast multi-GPU training and GPU/CPU translation with the use of state-of-the-art NMT architectures (deep RNN and Transformer). It supports training and translation of a number of popular NMT models. Underneath the NMT API lurkes a quite mature deep learning engine with no external dependencies other than boost and Nvidia's CUDA. Google is also developing its own Text-to-Text Transfer Transformer named T5 [3], which can convert any language problem into a text-to-text format. T5 is an extremely large new neural network model that is trained on a mixture of unlabeled text and labeled data from popular natural language processing tasks, then fine-tuned individually for each of the tasks that they authors aim to solve. The most obvious new idea behind this work is that it is a text-to-text model. During training, the model is asked to produce new text as an output even for training tasks that would normally be modeled as classification and regression tasks with much simpler kinds of output. One of the models we looked a bit deeper into was Fairseq [4] which is a sequence modeling toolkit that allows us to train custom models for translation, summarization, language modeling and other text generation tasks. It is based on PyTorch and features multi-GPU training on one machine or across multiple machines. It has multiple search algoritms like beam search, diverse beam serach and sampling implemented for fast generation on both CPU and GPU. It allows mixed precision training which helps with training on GPUs with less memory. Fairseq can be extended through five types of user-supplied plug-ins, which enable experimenting with new ideas while reusing existing components as much as possible. In the past Fairseq has been used in many applications, such as machine translation, language modeling, abstracitve document summarization, etc... It is meant for sequence modeling, it is scalable and suitable for many applications mentioned above.

Used method

After struggling to get a grip of how to run the Fairseq model we decided to switch over to a well documented open-source project called OpenNMT, which is designed to be research friendly. OpenNMT aims to provide a shared framework for developing and comparing open-source systems, while at the same time being efficient and accurate enough to be used in production contexts. The system is designed to be simple to use and easy to extend, while maintaining efficiency and state-of-the-art accuracy. The system was also additionally designed with two aims, the first being prioritization of training and test efficiency and the second to maintain model modularity and readability. At the heart of the project are libraries for training, using, and deploying neural machine translation models. OpenNMT has currently three main im-

plementations, the one we are using in our solution is the OpenNMT-py an OpenNMT-lua clone using PyTorch. When training GPU-based NMT models, memory size restrictions are the most common limiter of batch size, and thus directly impacting training time. OpenNMT has an external memory sharing system implemented that aggressively shares the internal buffers between clones. In the OpenNMT-py version it is implemented for both data loading to enable training on extremely large datasets that cannot fit into memory, and for back-propagation to reduce memory footprints during training. OpenNMT additionally supports multi-GPU training using data parallelism. Each GPU has a replica of the master parameters and processes independent batches during training phase. Two modes are available: synchronous and asynchronous training. The solution also explicitly separates training, optimization and different components of the model. Each module in the library is highly customizable and configurable with multiple ready-for-use features. The model supports different input types including models with discrete features models with non-sequential input such as tables, continuous data such as speech signals, and multi-dimensional data such as images. OpenNMT packages several additional tools, including reversible tokenizer, which can also perform Byte Pair Encoding (BPE), loading and exporting word embeddings, translation server which enables showcase results remotely and visualization tools for debugging or understanding, such as beam search visualization, profiler and TensorBoard logging.

Corpora selection

In choosing the appropriate corpus for the basis of our machine translator, we opted for the military corpus because it is a very interesting and specific topic. Even though the available military corpus that the teachers provided us contained a lot of data, we still needed more data to successfully build a specialised machine translator, so we searched for other bilingual documents. These were obtained mainly from the online sources of the Government and the Ministry of Defence of the Republic of Slovenia and the Slovenian Army. Some examples are the Defence White Book, the Defence Act and various agreements, protocols and laws. The bilingual documents then had to be aligned. We did so in SDL Trados Studio and converted them to an appropriate format (tmx or csv). We also included a large military termbase even though it cannot be used the same way as in CAT tools. This way, we obtained a sufficient amount of data to build and train the machine translator.

Work Process

With OpenNLP chosen as our translation network we had set the next goal of training our model on a smaller subset of corpora before building the full model.

We had decided to use Google Colab as our platform for training the model. Our data is primarily focused on multiple military documents issued by the Slovenian military. These documents have a Slovene and English translations and make together about four thousand sentences. All the documents first had to be converted into the right format, meaning aligning English and Slovene sentences line by line in the separate documents. Since four thousand sentences will likely not be enough for producing an efficient model we decided to include other corpora as well. We first thought of incorporating the EU Parliament corpus, which at least vocabulary-wise might be similar to our own domain. Besides that we had decided to include corpora from other sources as well (OpenSubtitles, EMEA, DGT and ELRC). The exact data set although will be known as the training process continues.

Before training our model we had built the vocabulary through the OpenNMT functions. Using the EU Parl corpora, we were able to extract about 250 thousand words in our target language. The same vocabulary was then used for training our translation model.

For training we first trained our model on about 600 thousand sentences from the EU Parliament corpus. We first trained and tested our model only on the EU Parliament data, but then we included our own domain corpus in the testing as well. In the early stages the model was not performing too well, but giving it some time the model becomes more and more precise.

We have not yet performed any evaluation of the translation, but we have looked into the translations and they seem pretty good for the first training. Here are some of the translations from the subsection of the EU Parl data of the model trained on the different subsection of that same corpus.

English text	**Ground truth**	**translated**
Action taken on Parliament's resolutions: see Minutes	Nadaljnje obravnavanje resolucij Parlamenta: gl. zapisnik	Nadaljne obravnavanje resolucij Parlamenta: glej zapisnik
The debate is closed.	Ta razprava je končana.	Razprava je končana
There is now binge-drinking in Spain, which used to be a north-western European problem.	Zdaj se popivanje, ki je bilo težava severozahodne Evrope, dogaja v Španiji.	Zdaj se (unk) v Španiji, ki se uporabljajo v (unk) evropski problem.
The fact is that the solution lies somewhere in the middle.	Rešitev je zagotovo nekje vmes.	Dejstvo je, da je rešitev v (unk)

Table 1. Translations on the EU Parl corpus.

Here are some of the translations from the subsection of the military data of the model trained on the EU Parl corpus.

English text	**Ground truth**	**translated**
Action taken on Parliament's resolutions: see Minutes	sprejeli bomo odločitve o partnerstvu z zavezniki glede urjenja na višji ravni .	Sprejeti bodo odločitve o partnerstvu z (unk) (unk) (unk)
a reduction in the military threat and the consequent abandonment of the concept of total defence	zmanjšanje vojaške ogroženosti in kot posledica tega opustitev koncepta totalne obrambe,	(unk) v vojaški grožnje in (unk) (unk) koncept celotne zaščite in (unk)
Slovenia must seek to formulate a social policy that guarantees social security for all its inhabitants.	RS mora oblikovati takšno socialno politiko, ki bo zagotavljala socialno varnost vsem prebivalcem RS.	(unk) si mora prizadevati za oblikovanje socialne politike, ki zagotavlja socialno varnost za vse prebivalce (unk)
replies provided to domestic mass media	odgovori na vprašanja domačih množičnih občil	Odgovor na domače množično medije.

Table 2. Translations on the military corpus.

From looking at the predictions it is pretty evident that the model is struggling more with the military corpus. That is of course expected since military corpora was not used in the actual training, but here we can see how important is choosing the right data for the domain we are trying to translate. Since our domain data is split into different documents we will combine all of them and randomize the sentences so we reduce the bias when we eventually split the data into train and test section. The whole corpora is around for thousand sentences out of which about 3 to 3,5 thousand of it will be used for training.

our goal is now to create successful translation model incorporating mentioned corpora including corpora from our own domain. After training the model we will evaluate our results using some of the methods that will be described in the next section.

Evaluation

For the evaluation of the results we are planning to use multiple algorithms for evaluating the translation quality. We are primarily looking at following:

- **BLEU** (Bilingual Evaluation Understudy) - standard algorithm for evaluating the machine translations against the human translations. It was one of the first metrics to claim a high correlation with human judgements of quality, and is still one of the most popular automated and inexpensive metrics. The central idea behind BLEU is that "the closer a machine translation is to a professional human translation, the better it is". BLEU's output is a value between 0 and 1, and indicates similarity between candidate and reference texts. Scores are generally calculated for individual sentences, by comparing them with a set of good quality reference translations. To estimate overall quality, scores are then averaged over the whole test dataset.
- **TAUS DQF** (TAUS Dynamic Quality Framework) - the TAUS Linguistic Quality Evaluation scorecard uses four error categories (accuracy, linguistic, terminology and style) and four levels of severity for each error category.This method of evaluation is not automatic but it is very thorough. Each combination of error category/severity is assigned a number of penalty points (the weight of the error type/severity combination). To pass this review, the translation cannot have more penalty points than an error threshold (normalized per/to 1,000 words).
- **WER** (Word error rate) - a common metric used to compare the accuracy of the translations produced by machine translation systems. The WER calculation is based on a measurement called the "Levenshtein distance." and is computed by dividing the number of errors(substitutions, deletions and insertions) by the total words. Lower WER often indicates higher accuracy, however, it should be used in combination with other metrics when determining the system accuracy.

Regardless of our focus on the above-mentioned metrics, we might as well use others like METEOR, ROUGE, TER, NIST, and RIBES.

One of the interesting experiments that we might tackle at the end would also be using the Google's translator and comparing our results to it. Even though we don't expect to achieve results better than google's translator, we believe that our model will give good results for the military domain, that we are focusing on.

Conclusion

As a part of the second assignment we have successfully implemented a translation model on a small subsection of the corpora we are planning to use. Our next goal is to include

other corpora including corpora from our domain to produce a more precise translation model for military related documents. After the translation will use some of the automatic translation quality metrics that we have discussed earlier.

References

[1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. 2019.

[2] 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*, Melbourne, Australia, 2018.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 1(3), 2019.

[4] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *NAACL*, 2019.