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

Automatic language translation (joint work with Faculty of Arts)

Marjan Bogataj, David Peklenik Urbanč, Aljaž Dobnik, Ilda Šećkanović, Saša Grahovac Fabbri, Jan Gajski

Abstract

Advisors: doc. dr. Slavko Žitnik, Mojca Brglez, Špela Vintar

Introduction

The idea behind the project is to create a natural language processing model for translating from the English language to Slovene and vice-versa. For of this task, we selected a neural translation framework that was trained with pre-selected general corpora in order for the model to learn basic language translations. We aim to evaluate how well the model learnt and translated provided texts with manual evaluation.

Related work

Due to the increasing use of machine translators in the translation process, there is more and more research dedicated to evaluating the quality of machine translation (MT). Evaluation of MT can be done either automatically, where computer metrics such as BLEU, GTM, METEOR compare MT with human translation, or manually, where a variety of assessment methods are used – one can evaluate adequacy, fluency, terminology, certain types of errors, or compare more translations of one text (Kuzman 2019 [1], 23–27) ... Bordon (2021) [2] analysed the end-user comprehensibility of unaudited machine-translated texts. In his questionnaire, he included texts translated by Google Translate and eTranslation from English into Slovene. Participants did not have access to the original text and saw translations in two types of contexts – with either an image or without. The results of the survey showed that the average comprehension rate was at 59%, whereby the eTranslation proved to be better than Google Translate. Vintar (2018) [3] dealt with the evaluation and comparison of a statistical phrase-based and neural model of Google Translate. She compared the translations between English and Slovene and focused on the terminology of karstology. The analysis showed that the neural translator generally translated better in both directions, but when translating terminology, it was better only in the direction from English to Slovene and not vice versa – from Slovene to English the statistical model was more successful. Kuzman (2019) [1] compared Google Translate and two of her models, which were specialized in translating literary texts from English into Slovene. The results of automatic and manual evaluation methods, except the fluency method, showed that a model tailored to a specific author was more successful than a model that was trained on a bigger corpus of different literary works. However, the best results with all evaluation methods were achieved by Google Translate.

Human evaluation of neural machine translated texts

Despite very accurate machine evaluation of machine translated texts, human judgment is essential for designing effective evaluation systems and interpreting the scores they provide. Human input is crucial when it comes to improving MT evaluation systems since human analysis often serves as a framework for the creation of such tools. Many human evaluation methods of machine texts exist. Apart from the most basic technique, where adequacy and fluency are evaluated, we can also compare several translations, decide to check only determined error types, evaluate reading comprehension of machine translated texts, focus only on determined phrases used in translations (Callison-Burch et al., cited in Kuzman, 2019) [1], or focus on terminology. In the present day, the most commonly used method, developed by TAUS, is used to evaluate neural-machine translated texts through adequacy and fluency. Translations are rated by level from 1 to 4, whereby 1 is the lowest level of adequacy/fluency and

4 the highest. The evaluators first read the original text and then the translation. The TAUS method requires evaluators to read and evaluate the same text. At this stage we have decided to evaluate our texts via adequacy and fluency, while at a later stage, when the neural machine translation system will be processing more specific texts, our evaluation will also include the evaluation of terminology. We hope to show how the most common inadequacies, realized by the neural machine translation model, can be addressed and how this can contribute to developing a new and better system for neural machine translation.

Data

At this project stage, the basic translator was trained exclusively using the Europarl parallel corpus. The corpus is a collection of parallel texts in 21 European languages from the proceedings of the European Parliament, dating from between 1996 and 2011 (Koehn 2005) [4]. For this project, only the Slovene-English parallel subcorpus was used. It consists of 623.490 sentences which further contain around 13 million Slovene and over 15 million English words. This bilingual subcorpus was downloaded in sentence aligned format (i. e. two monolingual text files, where each line represents one sentence). For our selected translation framework to utilize the corpus correctly, a space was added before and after all special characters if it was not already present (for example ",.-*?!'&). The corpus was then segmented into three separate parts:

- Training data - used for training the model with the translation framework, contains most of the corpus data
- Validation data - used for the evaluation of a model in the process of training, it is used to correct the model, contains about 6500 sentences
- Test data - used for evaluating the trained model, contains 7000 sentences

Methods

The framework chosen for this task was OpenNMT-py [6], developed by the University of Harvard and SYSTRAN and is currently maintained by SYSTRAN and Ubiqus.
OpenNMT-py is an open-source neural machine translation system designed to be research-friendly and can be used for natural language translation. It also uses PyTorch, which is widely used across machine translation frameworks. OpenNMT-py was chosen for its ease of use and well-written documentation.

Algorithms

OpenNMT-py requires that the corpus contains each language in its file and every sentence in its line, with special characters separated from other words with a space. The starting corpus Europarl already meets most of these criteria, except the last one, which is done by *convert_korpus.py*. With the corrected corpus, a vocabulary was built using the OpenNMT-py command *onmt_build_vocab* on the first 10.000 sentences for English and Slovene corpora. Next, the translation model was trained using both the corrected Europarl corpus and the vocabulary. The parameters for training the model are the location of English and Slovene vocabularies, the location of the *train* and *validation* files for both languages, number of GPU's to train on, location of the saved model, number of steps (iterations) the model will train for, and how often the model will be evaluated. These parameters are stored in *conf.yaml* file, which is used with OpenNMT-py command *onmt_train* to train the model. Finally, the trained model can be used to translate from one language to the other. This is done with the OpenNMT-py command *onmt_translate* and the pre-generated *test* corpus. The translation is also being evaluated at the same time, with higher scores signifying better translations.

Results

The model was trained using three different settings for the number of steps (a setting, which directly impacts how long the training will take) and two different frequencies of model evaluation during training. This was done for both translation models (English to Slovene and vice-versa). In the first scenario, the English-Slovene model was trained with only 1000 steps and evaluated every 500th step (meaning twice for this scenario). The results were poor since most of the sentences for translation comprised of unknown words(tagged as $< unk >$). Most of the translations proved to be very similar and contained many repetitions. Example of the first scenario:
SENT 6009: ['This', 'is', 'the', 'proof', 'that', 'fiscal', 'stability', 'leads', 'to', 'growth', 'and', 'employment', '.']
PRED 6009: To je zelo pomembno , da je $< unk >$.
PRED SCORE: -15.0615

Note: The sentence "To je zelo pomembno" was being repeated in many other translations.

The second scenario had the same validation frequency (every 500 iterations), but the number of steps for training was increased to 3000. This yielded much better results, since there were fewer $< unk >$ words, with actual sentences being formed. However, the translations themselves were deficient - the sentences didn't make much sense. It seemed as if the model was purely guessing the correlations between both languages, although some words were translated correctly. Example of the second scenario:
SENT 6009: ['This', 'is', 'the', 'proof', 'that', 'fiscal', 'stability', 'leads', 'to', 'growth', 'and', 'employment', '.']
PRED 6009: To je tisto , ki je $< unk >$, da je $< unk >$ $< unk >$ in $< unk >$.
PRED SCORE: -19.7641

In the third scenario we further increased the number of

steps to 6000. This improved the readability of the sentences and greatly improved the accuracy of translations. The unknown $< unk >$ words appeared far less frequently, they were not present in every translation anymore, and some short sentences were even translated correctly. Example of the third scenario:
SENT 6009: ['This', 'is', 'the', 'proof', 'that', 'fiscal', 'stability', 'leads', 'to', 'growth', 'and', 'employment', '.']
PRED 6009: To je dokaz , ki je $< unk > < unk > < unk >$ za rast in zaposlovanje .
PRED SCORE: -7.8747

The last test was done with the same number of steps as the third experiment (6000), but with the increased frequency of model validation (from every 500 iterations to every 250 iterations). This didn't change the resulting translations by much, although surprisingly, the average predicted score even worsened, which may indicate overfitting to the validation set.

Example of the last test:
SENT 6009: ['This', 'is', 'the', 'proof', 'that', 'fiscal', 'stability', 'leads', 'to', 'growth', 'and', 'employment', '.']
PRED 6009: To je dokaz , da je za rast in zaposlovanje .
PRED SCORE: -7.6077

The Slovene-English model behaved in a similar way, although there were fewer $< unk >$ words when the number of training steps was 1000. There were a lot of repetitions and the sentences didn't make sense. Example:
SENT 5336: ['Ljudi', ',', 'ki', 'bodo', 'skrbeli', 'za', 'kmetijski', 'sektor', ',', 'pa', 'tudi', 'za', 'področja', 'pred', 'in', 'za', 'njim', '.']
PRED 5336: I would like to thank the Commissioner , I would like to thank the Commissioner , I would like to thank the Commissioner .
PRED SCORE: -30.2082

The increase in the number of steps for training to 3000 had a positive impact on the quality of translations, similarly to the English-Slovene model, meaning the translations contained some correct words but the model couldn't correctly form a sentence. Example:
SENT 5336: ['Ljudi', ',', 'ki', 'bodo', 'skrbeli', 'za', 'kmetijski', 'sektor', ',', 'pa', 'tudi', 'za', 'področja', 'pred', 'in', 'za', 'njim', '.']
PRED 5336: People who will have a ¡unk¿ system , but we also have to ensure that we are in favour of a system and for you .
PRED SCORE: -34.6084

Similarly to the English-Slovene model, the increase of steps to 6000 greatly improved the readibility of the translations, the number of $< unk >$ decreased, and some shorter sentences were translated correctly. Example:
SENT 5336: ['Ljudi', ',', 'ki', 'bodo', 'skrbeli', 'za', 'kmetijski', 'sektor', ',', 'pa', 'tudi', 'za', 'področja', 'pred', 'in', 'za', 'njim', '.']
PRED 5336: People who are going to be ¡unk¿ for agricultural sector , and also for them .
PRED SCORE: -13.7145

Evaluation

Evidently, the longer the model is trained, the better it understands the correlation between languages and therefore produces improved translations. It is also important to note that the vocabulary was built on only the first 10.000 sentences from a 600.000 sentence corpus, but the number of unknown words still decreased rapidly after increasing the number of training steps for the model from 1000 to 3000. The evaluation frequency isn't as important, but it can lead to overfitting to the validation corpus, which worsens the translational accuracy. This can be observed with the average prediction score, where the example with fewer validations (Example 3) had a score of -0.8293, and Example 4 with doubled validation rate had a score of -0.9301 . There was little difference between the training process between the English-Slovene and the Slovene-English model. They both showed similar results relating to the number of training steps.

Future work

The model is currently trained using only the Europarl dataset for the general translation model, and it is planned to expand the general corpus to not only include Europarl but several other corpora to improve the quality of the vocabulary and general language comprehension of the model. Furthermore, the model will be trained on specialized corpora in order to allow the model to be able to perform better at specific topics.

References

[1] Kuzman, Taja. 2019. »Nevronsko strojno prevajanje literarnih besedil iz angleščine v slovenščino« [Neural Machine Translation of Literary Texts from English to Slovene]. Master's thesis. University of Ljubljana.

[2] Bordon, David. 2021. »Razumevati nevronščino: kako si ljudje razlagamo jezik strojnih prevajalnikov« [Comprehending the neural language: How people understand the language of machine translation engines]. Master's thesis. University of Ljubljana.

[3] Vintar, Špela. 2018. »Terminology Translation Accuracy in Statistical versus Neural MT: An Evaluation for the English-Slovene Language Pair«. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). http://lrec-conf.org/workshops/lrec2018/W19/pdf/7_W19.pdf.

[4] Koehn, Philipp. 2005. »Europarl: A Parallel Corpus for Statistical Machine Translation.« MT Summit 2005. https://www.statmt.org/europarl/.

[5] Escribe, Marie. 2016: Human Evaluation of Neural Machine Translation: The Case of Deep Learning, Guildhall School of Business and Law London Metropolitan University. London. Available via: https://www.aclweb.org/anthology/W19-8705.pdf

[6] OpenNMT-py, https://github.com/OpenNMT/OpenNMT-py/. Last accessed 1 May 2021