Persian - Romanian Machine Translation

1st Semester of 2021-2022

Ghadamiyan Lida
lida.ghadamiyan@s.unibuc.ro

Oanea Șmit Andrei
smit.oanea@s.unibuc.ro

Ouatu Bogdan Ioan
bogdan.ouatu@s.unibuc.ro

Abstract

The main purpose of this project was to create a machine translation model that would be able to directly translate sentences from Farsi to Romanian. This pair of languages has not been approached in the literature successfully. Classically, large parallel data sets (few GB) and substantial computing power are required to train such a combination of languages. Even though we did not possess these resources, we achieved results comparable to SOTA on pairs of similar languages. We used the transfer learning method in order to reach the BLEU score of 35.55 with 90 Mb of data and 10 hours of training on a single GPU.

1 Introduction

Our goal for this project was to create a machine translation system for Persian and Romanian. We chose this pair of languages due to multiple reasons, as not being approached before, the resemblance of the languages (both are indo-european, having a lot of similar words) and the affinity of our team to this task.

A similar approach with ours was used in (Tom Kocmi and Ondrej Bojar, 2018) in which they trained Transformer sequence-to-sequence model (Vaswani et al., 2017) on a high resource language pair for 140 hours after which they stopped training and continued it on a low resource pair continuing with the same hyperparameters. Others such as (Zoph et al. 2016) and (Nguyen and Chiang 2017) propose additional constraints such as the low-resource language pairs have to be related or at least one language to be shared between them.

We propose that the initial training can be done successfully without any hyperparameter sharing and, as in (Tom Kocmi and Ondrej Bojar, 2018), with no restrictions on language relatedness. We obtain better results than presented in the previously cited papers using a preexisting trained model from multiple languages to English described in the Methods section.

Our contribution consists in:

- Bogdan did the initial research and organised us by choosing the approaches and coordinating the training sessions. He also created a data set from the initial corpora, by analyzing and preparing the input for the model, tuned the model and documented his steps for the final paper.
- Smit found the open-source datasets and created the main corpora we used. He searched for available translation APIs for python in order to do the pivot language approach, and he also documented his contributions for the final paper.
- Lida came with the idea of using subtitles to generate a parallel corpora and created the small data set. She also did the data preprocessing and visualization, used the model (without tuning), computed the BLEU score and wrote the final paper.

2 Approach

The project can be foud at: https://github.com/Ghadamiyan/Farsi-Romanian-Machine-Translation

2.1 Used Datasets

We combined a number of three open-source datasets from https://opus.nlpl.eu/: Tanzil [1], GNOME [2] and TED2020[3] in order to obtain a more complete dataset that we named Clean Corpora. We also tried another dataset obtained from the subtitles of three movies: Encanto, Hachiko and Spiderman: No way home. The training set has 300000 Romanian-Farsi sentences and has 3 columns in addition to the unique identifier column:

ro - which incorporates sentences in Romanian language, fa - which retains the translated sentences in Farsi language, source - which keeps the root where the information was taken from. The test dataset has 3300 samples. The process of tokenization was made by using AutoTokenizer from transformers which splits the sentences to subwords and it is specially trained for the Helsinki (NLP) model that we used.

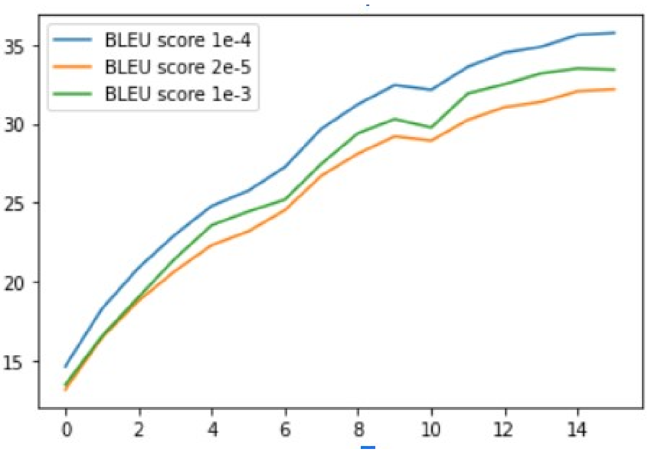
2.2 Datasets obtained from paralel subtitle pairs

We used materials that were translated in multiple languages, containing the ones we needed. The downside of this situation is that the datasets are small for this task and it contains a lot of archaic words used in the Bible. As a solution to this problem, we developed a method to collect suitable data. We used available subtitles for recent movies, such as Encanto and Spiderman. The advantage of using subtitles is that each line has a time stamp that we used to synchronize the translations. We made sure the subtitles are synchronized by reading some of the first lines. Another advantage of using this data is the diversity of the vocabulary, as the movies imitates the real life conversations, and also the lack of grammatical errors. In order to create the datasets we converted the srt files to txt and then extracted the sentences and their starting time stamp. We did that for both languages and then merged them using the time attribute. It did not take us a long time to obtain the data from a single movie, so it is a good approach to use in order to collect data.

2.3 Translation using Microsoft Azure API

We used the Microsoft Azure Translator API as a baseline method that we can compare our results with. We tried both direct persian-romanian translation and pivot translation through the english language. We also suspect that the Azure Translator service uses indirect translation through the english language, so we compute the BLEU score between the direct translation and the indirect one. The result is 99.44, which solidifies our suspicion. Also, the BLUE score for the indirect translation is very similar but still slightly higher than the direct translation's score (12.34 compared to 12.33), which would be counter-intuitive for a not-pivot translation, as the indirect methods tend to perform slightly worse.

Figure 1: BLEU score during training plotted against learning rate



2.4 OpenNMT

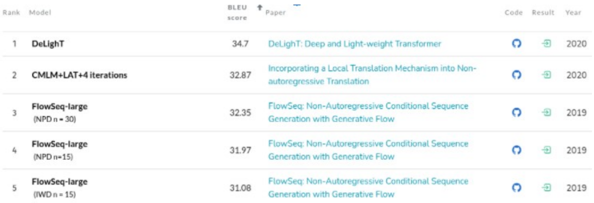
We used the Opus-MT-mul-to-en model [7] based on the transformer architecture implemented in MarianNMT[8]. The transformer model is based on an encoder and a decoder which can process up to 128 tokens. We found that it's a good balance between memory consumption and dataset coverage, >90The model is trained with a SentencePiece tokenization[9], a sub-word tokenization technique.

2.5 Hyperparameters

We used a 1e-3, 1e-4 and 2e-5 learning rate as in Fig. 1. We used gradient accumulation with 5 steps to speed up training.

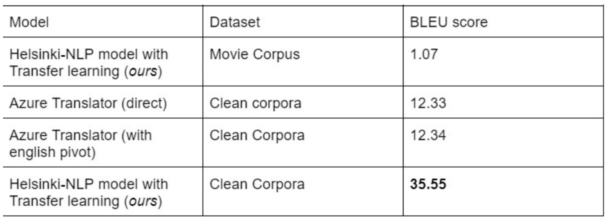
3 Testing and Evaluation of Results

English - Romanian The BLEU score from literature for this language pair reached 34.7 in 2020. (paperswithcode.com benchmark scores for English-Romanian)



Rank	Model	BLEU score	Paper	Code	Result	Year
1	DeLighT	34.7	DeLighT: Deep and Light-weight Transformer	○	⇥	2020
2	CMLM+LAT+4 iterations	32.87	Incorporating a Local Translation Mechanism into Non-autoregressive Translation	○	⇥	2020
3	FlowSeq-large (NPD n = 30)	32.35	FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow	○	⇥	2019
4	FlowSeq-large (NPD n=15)	31.97	FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow	○	⇥	2019
5	FlowSeq-large (IWD n = 15)	31.08	FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow	○	⇥	2019

4 Results



Model	Dataset	BLEU score
Helsinki-NLP model with Transfer learning (*ours*)	Movie Corpus	1.07
Azure Translator (direct)	Clean corpora	12.33
Azure Translator (with english pivot)	Clean Corpora	12.34
Helsinki-NLP model with Transfer learning (*ours*)	Clean Corpora	**35.55**

5 Conclusions and Future Work

We showed that using transfer learning on low-to-medium resources with little computing power yields results comparable with SOTA.

We enjoyed doing this project because it was the first time we are doing machine translation and it was interesting to see how the model translated different sentences.

Bibliography

[1]Tanzil Corpus: J. Tiedemann, 2012, Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012). This is a collection of Quran translations compiled by the Tanzil projectTerms of UseThe translations provided at this page are for non-commercial purposes only. If used otherwise, you need to obtain necessary permission from the translator or the publisher.If you are using more than three of the following translations in a website or application, we require you to put a link back to this page to make sure that subsequent users have access to the latest updates. (http://opus.nlpl.eu)

[2] J. Tiedemann, 2012, Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)

A parallel corpus of GNOME localization files. Source: https://l10n.gnome.org

[3] J. Tiedemann, 2012, Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)

This dataset contains a crawl of nearly 4000 TED and TED-X transcripts from July 2020. The transcripts have been translated by a global community of volunteers to more than 100 languages. The parallel corpus is available from https://www.ted.com/participate/translate

[4] J. Tiedemann, 2012, Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)

A parallel corpus extracted from the European Parliament web site by Philipp Koehn (University of Edinburgh). The main intended use is to aid statistical machine translation research.More information can be found at http://www.statmt.org/europarl/.

[5] J. Tiedemann, 2012, Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)

A parallel corpus of TED talk subtitles provided by CASMACAT: http://www.casmacat.eu/corpus/ted2013.html. The files are originally provided by https://wit3.fbk.eu.

[6] Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong and Paco Guzman, WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia, arXiv, July 11 2019.

Parallel corpora from Wikimedia compiled by Facebook Research The data is released under the Creative Commons Attribution-ShareAlike

[7] OPUS-MT — Jorg Tiedemann and Santhosh Thottingal Building open translation services for the World

[8] Junczys-Dowmunt et. al Marian: Fast Neural Machine Translation in C++

[9] Taku Kudo, John Richardson - SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing

[10] Tom Kocmi Ondˇrej Bojar - Trivial Transfer Learning for Low-Resource Neural Machine Translation https://aclanthology.org/W18-6325.pdf

[11] https://ytsubtitles.com/