Triangular Machine Translation

1st Semester of 2021-2022

Anca Sotir
anca.sotir
@s.unibuc.ro

Irina Chitu
irina.chitu
@s.unibuc.ro

Teodor Marchitan
teodor.marchitan
@s.unibuc.ro

Abstract

Machine Translation is generally considered to be a data-hungry task, especially when neural networks are involved. Even nowadays there are many low-resource pairs of languages on which these models do not perform well due to their nature. A very natural strategy to help minimize this setback is to introduce an intermediate language, one with significantly greater resources. The classic example is using English as that middle language, due to the fact that it is the most widely used. This idea is the basis of Triangular Machine Translation, as it is also hinted in the name of this concept. In this paper we aim to compare the results of multiple experiments for the translation from Bulgarian, Greek and Serbian to Romanian.

Key words: Machine Translation, Triangular MT, SETimes, Bulgarian-Romanian, Greek-Romanian, Serbian-Romanian

1 Introduction

1.1 Task description and motivation

Triangular MT is a strategy which can prove itself useful especially in cases of lower-resources language pairs. The main idea is that instead of directly translating from X to Y, we introduce a middle-language P (called pivot) and first translate from X to P and then from P to Y. The pivot can usually be chosen as a richer language, such as English, in order to combat the setback of fewer resources for the initial language pair. Another remark would be that we cannot always find pretrained models for certain pairs of languages (X/Y), but the probability of finding pretrained X-to-English and English-to-Y models is far more likely. These pretrained models could potentially improve translation results, provided the fact that they have seen much more training data.

A shared task on Triangular MT was introduced in (WMT21) in order to bring more attention to combining the direct X/Y and the pivot X/English+English/Y approaches. While the initial task was to improve Russian-to-Chinese translation, we decided to tackle pairs of languages which are closer to us as native speakers of Romanian. Thus, the final theme of our project is **Triangular MT** on the following language pairs: **Bulgarian/Romanian** (bg-ro), **Greek/Romanian** (el-ro) and **Serbian/Romanian** (sr-ro).

1.2 The contribution of each member

- Anca Sotir has covered the direct translation from Bulgarian, Greek and respectively Serbian to Romanian.
- Irina Chitu has covered the single pivot approach, using English, for all three language pairs mentioned before.
- Teodor Marchitan has covered the double pivot approach building a translation pipeline X $\rightarrow$ English $\rightarrow$ Macedonian $\rightarrow$ Romanian (where X is, of course, Bulgarian, Greek and Serbian).

1.3 A summary of our approach

As it can be seen in the contribution per member, we have split the task into three separate experiments. Our final aim for this project is to compare their results.

1. **Direct approach:** we did not find any pretrained models directly from bg, el or sr to ro. Thus, we decided that for this experiment we will use the `fairseq` framework in order to train a model for each language pair (bg-ro, el-ro, sr-ro).
2. **Single Pivot approach:** We are introducing a popular language (such as English) as a pivot

to act as a bridge between the chosen languages. Not only did we find intermediary pre-trained models, but the results were highly favourable from both manual and automatic evaluation.

3. **Double Pivot approach:** We'd like to further enhance the previously mentioned method by adding another pivot. However, this time the focus lies on its similarity with the chosen language.

1.4 Related work

To have a good idea about the task, we looked over the papers submitted at WMT21 for the triangular MT problem. Many of the researched papers were using the idea of transformers introduced by (Vaswani et al., 2017), an encoder-decoder architecture based on a self-attention mechanism. Apart from this, we also discovered some new ideas such as one related to data augmentation, which says that we can enrich our corpus for languages X and Y by using a pivot language Z (Park et al., 2021). We can translate the sentences in language Z, from the corpus corresponding to pair X-Z, into the language Y, obtaining such extra data for the pair X-Y. A similar technique can be used for data from pair Y-Z. Another approach worth mentioning is using the idea of transfer learning in triangular MT. Precisely, after training a transformer for X-Z and one for Z-Y, instead of sequencing these models, we will take the encoder from the former model and the decoder from the latter. These components are then used to initialize the weights for a direct transformer X-Y (Mhaskar and Bhattacharyya, 2021).

1.5 What we learned. What else we want to learn related to this project

Anca Sotir: After understanding what triangular MT does, I realised it is very natural, because in a real world situation if two people with different native languages want to communicate, they will try to find a common ground. This idea made me wonder about how many other scientific challenges have solutions in our day to day life. Regarding what I want to learn more about machine translation, the concept of multilingual model got my attention. It can be trained on multiple pairs of languages and afterwards generate translations for different sources and targets including pairs that were not used for training.

Irina Chitu: I believe that having a final project at the end of a course is the best way to sum up all the learnt information and to understand how to apply it. Moreover, having the freedom to explore pre-trained models was a great plus since in the future my focus will not be solely on research. I'd rather know of the existence of several concepts and where to find them than learn them by heart.
What I enjoyed the most while working on this project was seeing how the results improved (in the beginning our translations were pure nonsense)

Teodor Marchitan: The main idea of the project was very interesting and I discovered a lot of new tools and techniques. It was satisfying to see positive results and succesfull experiments. But there were also some disadvantages such as: the necessity of a very big dataset in order for the models to perform well, hardware limitations and long training times and very different frameworks API to work with. The most annoying thing during the project was that every framework that we wanted to use was using a totally different type of input data, the usage of the framework was totally different. So because of this, trying different experiments just with some minor changes to the configurations was pretty inconvenient as it would take some time to understand what to change and how to change.

2 Approach

Note: the code for our experiments can be found on github at this link.

We have chosen our data from SETIMES (Ljubešić), a parallel news corpus containing 10 languages, including all the languages we need for our experiments: Romanian, Bulgarian, Greek, Serbian, English and Macedonian. For each pair of languages we have around 200 000 sentences.

For a fair comparison between experiments, the first 10 000 samples were used for training, the next 2 500 for validation and the following 2 500 were reserved for testing.

2.1 Direct Translation

For each language pair (bg-ro, el-ro, sr-ro) we have used the same approach, which is based on the second laboratory of our Machine Translation Course.

As a preprocessing step, we checked for any empty or too long sentences, and also for pairs of

<source_sent, target_sent> which had a high length ratio (which means their lengths are significantly different). Virtually no such entries were found for this dataset.

Subword segmentation was also performed using `SentencePiece`. We found that using a larger vocabulary improved the BLEU score on validation data.

For training we used the `fairseq` framework. The model had the transformer architecture and multiple combinations of hyperparameters were tried.

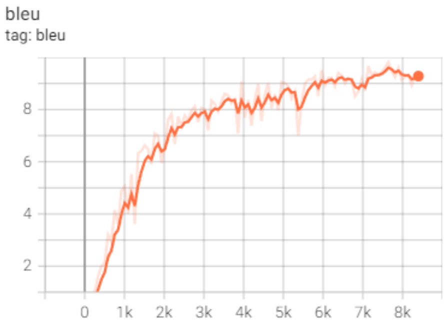

Figure 1: Validation BLEU during training (bg-ro).

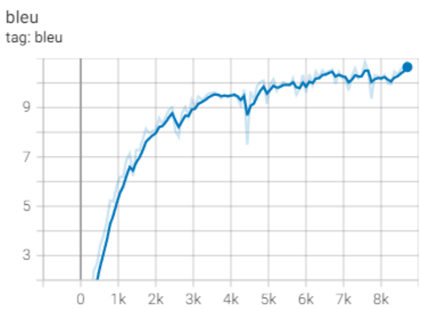

Figure 2: Validation BLEU during training (el-ro).

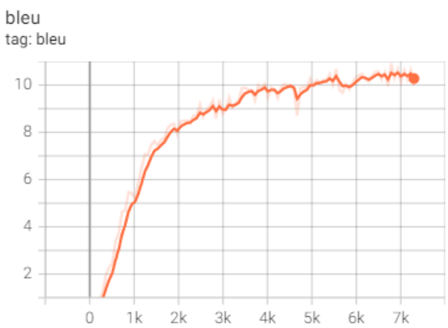

Figure 3: Validation BLEU during training (sr-ro).

Four our data setup, increasing the model complexity did not improve results, but allowing the model to train for more epochs did. In total, the model had around 815 000 trainable parameters and it was trained for 100 epochs. For Bulgarian, the validation BLEU score reaches a value between 9 and 10 and for Greek and Serbian between 10 and 11.

2.2 Single Pivot

The single-pivot approach is a great work-around for languages with little to no datasets and no existing pre-trained machine translation models. Moreover, this method opens up a bright path when choosing a well-known language as a pivot.

The implementation is based on several PyTorch (PyTorch) libraries from `Hugging Face` (Face), a NLP-focused startup with many state-of-the-art results.

Using the datasets library we import the SETIMES above mentioned language pairs formatted as a list of jsons (e.g. {<source_lang>: <source_text>, <target_lang>:<target_text>}).

No direct pre-trained models exist for the chosen language pairs. However, by choosing a popular language such as English for the pivot we can now import several models and their tokenizers from the transformers library. The name comes from a type of neural network architecture with the same name with the purpose of transforming one sequence into another. This is done with the help of two parts: Encoder and Decoder(Maxime). The former produces a representation for the source sequence, while the latter uses the representation in order to generate the target sequence. Our chosen model is a transformer encoder-decoder with 6 layers in each component. It is similar to BART (Mike Lewis) with a few minor modifications .

After loading the necessary models, we compute the translation in the following manner: for a given input in language A (where A is Bulgarian, Greek or Serbian), we tokenize it and generate a translation in English (the pivot) using the corresponding model and tokenizer; afterwards, the latest output becomes the new input for the en-ro model; again, the input is tokenized and then translated.

2.3 Double Pivot

In this approach we are investigating whether or not an additional pivot will improve the results. We are now going through two pivots before reaching our final translation: first English, then Macedonian. If we previously chose English for its popularity, Macedonian was chosen due to its similarity to our source languages. We are relying on the first pivot

Method	BLEU score
`bg-en-ro`	36.4931
`grk-en-ro`	35.5486
`sr-en-ro`	25.8353
`bg-en-mk-ro`	23.1548
`grk-en-mk-ro`	20.8195
`sr-en-mk-ro`	23.8915

Table 1: BLEU scores for the pivots approaches

for a raw translation and on the second one for a meaningful translation. The order of the pivots was influenced as well by the variety of existing pre-trained models.

The implementation is built upon the previous one, therefore we make use of the Hugging Face libraries for the available translations. For the missing ones, we tried two approaches. Our first idea was to explore the `fairseq` framework (Myle Ott), but unfortunately our dataset was too big for our hardware and just by taking a small subset of just 10k samples, after 10 epochs the results were extremely poor. Our second idea was more around the similarity concept. For example, for our missing pair Serbian-English we looked for a similar language to Serbian which already had a pre-trained model and we found Ukrainian (ezglot). Then, we fine-tuned (Kumar) the existing uk-en model for another ten epochs on our sr-en dataset. We approached the missing mk-ro model in the same way and the BLEU score increased from 9.38 (first epoch) to 23.66 (last epoch) .

3 Conclusions and Future Work

For the pivot approaches we computed the BLEU scores, which can be seen in Table 1.

Figures 4, 5 and 6 display a manual evaluation between the different approaches. The texts are what the output translations.

Bulgarian - Romanian	
True translation	După aceasta a fost simulată explozia şi distrugerea unui pod.
Direct Translation	După ce a fost împărţit de a fost împărţirea de distrugere şi distrugerea şi distrugerea de distrugerea şi distrugerea de distrugerea.
1 pivot	Apoi s-a făcut o simulare a exploziei şi distrugerii unui pod.
2 pivots	După aceasta, situaţia s-a îmbunătăţit şi s-a îmbunătăţit într-o singură locaţie.

Figure 4: Bulgarian-Romanian translations

Greek - Romanian	
True translation	Am remarcat de asemenea lipsa corelării dintre proiectele finanţate chiar şi în acelaşi judeţ.
Direct Translation	Am reuşit să asigure de asemenea proiectele proiectelor dintre jumătatea proiectelor dintre jumătate.
1 pivot	De asemenea, am observat o lipsă de corelare între programele finanțate chiar și într-o țară.
2 pivots	De asemenea, am văzut lipsa unei comisii printre programele financiare şi în cadrul unei ţări.

Figure 5: Greek-Romanian translations

Serbian - Romanian	
True translation	Omologul azer al lui Videanu a numit semnarea memorandumului drept "un eveniment istoric".
Direct Translation	Videanu s-a concentrat asupra memorandumului de înţelegere asupra memorandumului de înţelegeree".
1 pivot	Omologul azer al lui Vidanu a numit semnarea memorandumului "evenimente istorice."
2 pivots	Colegul Azerbejjanski de la Videanu a făcut apel la "insultarea tradiţiilor istorice".

Figure 6: Serbian-Romanian translations

3.1 Direct Translation

Some general observations based on the models' results would be that repeat words or similar word sequences too many times in a row. These direct translations have grammatical errors and are not very coherent. Some key, easier words might correctly appear in the translation, but the overall meaning of the target text is not the same (this might be the case only for very basic sentences).

These are some ideas that could potentially improve this experiment's results:
- change the architecture of the model
- train on more data

3.2 Single Pivot

We saw that wisely choosing a pivot brings satisfactory results not only to the evaluation but also to the overall process. When either time or resources are lacking, pre-trained models are the solution. A possible improvement to this approach is to change the pivot to some other languages and compare the results.

3.3 Double Pivot

Even though the BLEU score did not improve comparing to the second approach, this method comes with its own benefits. In case we decide that a certain language would perfectly fit as a pivot but we are lacking datasets to and from that pivot, by adding a second pivot we increase the chances of finding a pre-trained model. Some improvements can be made here as well. For example, in the future, we could experiment with either changing the order of the pivots (in our case bringing closer in the translation process the similar languages) or with the combination of pivots.

As a final remark, an overall improvement of this project would be attempting to combine the direct translation with the single or double pivot approach, and we would need to further study this problem in order to achieve this. It would be interesting to see if they provide the best results when combined.

References

ezglot. Most similar languages.

Hugging Face. Hugging face.

Sravan Kumar. How to fine-tune pre-trained translation model.

Nikola Ljubešić. Setimes.

Maxime. What is a transformer?

Shivam Mhaskar and Pushpak Bhattacharyya. 2021. Pivot based transfer learning for neural machine translation: Cfilt iitb@ wmt 2021 triangular mt. In *Proceedings of the Sixth Conference on Machine Translation*, pages 336–340.

Naman Goyal Marjan Ghazvininejad Abdelrahman Mohamed Omer Levy Ves Stoyanov Luke Zettlemoyer Mike Lewis, Yinhan Liu. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Alexei Baevski Angela Fan Sam Gross Myle Ott, Sergey Edunov. Fairseq: A fast, extensible toolkit for sequence modeling.

Jeonghyeok Park, Hyunjoong Kim, and Hyunchang Cho. 2021. Papago's submissions to the WMT21 triangular translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 341–346, Online. Association for Computational Linguistics.

PyTorch. Pytorch documentation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

WMT21. Shared task: Triangular mt: Using english to improve russian-to-chinese machine translation.