Machine Translation
Romanian-to-Romani Translation

1st Semester of 2024-2025

Zibileanu Sabin **Grigore Dragoș - Gabriel** **Vasilescu Costin - Tiberiu** **Handac Alexandru - Radu**
University of Bucharest

Abstract

We present the steps and the tools necessary to build a machine translation deep learning model for Romanian to Romani. Even though it is a low - resource language, the Romani language is a very popular one around the world and this project represents a starting point for future research and improvements on building machine translation models with this language.

How to use the project

1. in the jupyter notebooks available in the repository, all the important parts of the project can be found: data acquisition, tokenization, training, loading the data, different usage of the models etc.

2. knowing the fact that we are working with a low - resource language (Romani), we decided that it was best to upload the whole project for future research and why not, future usage.

3. the dataset is structured in two txt files: romani.txt and romanian.txt containing the acquired samples and also 2 tokenizer directories generated from creating a custom Romanian-to-Romani tokenizer with a base MarianMT Model and another directory containing an extended pre-trained tokenizer.

4. there are also 2 directories in the repository: "data" and "tokenizer", where you can find the code for obtaining the dataset and the tokenizer

5. if you want to reproduce the results, you can run the notebooks in the main branch.

1. **Data Acquisition**

   working with a low-resource language, we took into account the fact that we had to build a corpora from scratch almost. however, we found a corpus online with around 7900 samples from Romanian to Romani **from HelsinkiNLP** (Christodoulopoulos and Steedman, 2014), which to our knowledge is in the Carpathian Romani dialect.

   the available corpora was a great starting point, as it provided a considerable amount, but we wanted to add more data.

   to add more data, we scraped samples from **Glosbe** as it provided Carpathian Romani samples translated into Romanian.

   we chose the dialect Carpathian Romani as it is one that provides the most examples.

   because Glosbe is a dictionary website, we had to search for keywords first in order to scrape the resulted data. Those keywords can be: conjuctions, verbs, adverbs, prepositions etc.

   it is important to mention that some samples were added manually, because of the fact that certain keywords had multiple available phrases split into different pages and we could not load more pages with our scraper.

   next on, we manually gathered a set of samples from different dictionaries, conversational guides, and a Romani course support. We added those samples manually, due to the fact that we mostly chose phrases and also the dialectal mix between the already existing samples in Carpathian Romani and these samples in Kalderash Romani.

   a high mix of dialects between the samples could negatively affect the performance of the model.

2. **Tokenization**

   the Romanian-to-Romani tokenizer was created using SentencePiece, with a vocabulary size with respect to the size of the unique tokens in the dataset, which was around 10000.

after it was created, the tokenizer was loaded using MarianTokenizer (Junczys-Dowmunt et al., 2018)

the second tokenizer was an extended version of a pre-trained tokenizer from HelsinkiNLP: roa - en.

3. **Evaluation**

for a low resource language like Romani, human evaluation would be the best.

in our project, however, we present the results obtained with the BLEU metric (Papineni et al., 2002).

1 Introduction

The Romani language, although a low - resource one is, according to some estimates in the top 3% of the world's most spoken languages, containing multiple dialects: Carpathian, Kalderash, Balkan etc. Considering the fact that there are no available Romanian - Romani machine translation models (to our knowledge), we:

- explored multiple approaches for building a machine translation model for Romanian to Romani.

- initially used the MarianMT model (Junczys-Dowmunt et al., 2018) with two variations: one with a custom tokenizer and another with an extended pre-trained tokenizer.

- extended our experiments by fine-tuning two additional models: NLLB-200 (facebook/nllb-200-distilled-600M) and Llama-3.2-3B (unsloth/Llama-3.2-3B-Instruct-bnb-4bit) to evaluate their effectiveness.

- chose these approaches due to their different architectures: MarianMT is a well-structured Sequence2Sequence model, NLLB-200 is optimized for multilingual translation tasks, and Llama-3.2-3B is a large language model capable of handling generative translation tasks.

- to our knowledge, there are no papers covering this specific task with the models we explored.

- During this project, we learned:
  - more about data augmentation techniques for low-resource languages
  - how tokenization affects translation quality
  - different types of machine translation architectures
  - how to fine-tune various deep learning models for translation
  - how preprocessing techniques impact BLEU scores and model performance
  - challenges of creating a Romanian-Romani corpus

2 Approach

This section covers our approach, divided into detailed steps:

1. Data collection: We merged two datasets—one built from scratch using Glosbe, dictionaries, conversational guides, and a Romani course, and another from **HelsinkiNLP**. Duplicate samples were removed.

2. Data augmentation: Since Romani is a low-resource language, we applied:
   - **Back translation**: Romanian sentences were translated into English and back into Romanian.
   - **Sentence insertion**: A predefined token was added at the beginning of sentences to provide contextual clues.
   - **Random character insertion**: A random character was inserted into words to simulate typos (applied to 15% of the training set).

3. Model training:
   - **MarianMT**: Initially trained with custom and extended tokenizers.
   - **NLLB-200**: Fine-tuned for Romanian-Romani translation to evaluate performance on a model designed for multilingual tasks.
   - **Llama-3.2-3B**: Fine-tuned for generative translation to analyze its capabilities on this task.

4. Training details:

- MarianMT Training was conducted on Google Colab with a T4 GPU, duration ranged from 1 to 2 hours, depending on the model.
- NLLB-200 and Llama-3.2-3B were trained on Kaggle with 2xT4 GPU with training times of up to 10 hours.

5. Evaluation: Results were compared using BLEU scores.

MarianMT Results				
Data	**Model**	**BLEU**	**Epochs**	**Tokenizer**
Augmented	Base	1.969900	10	Custom
Augmented	Pre-trained	**6.726000**	10	Extended
Non-augmented	Base	0.823531	10	Custom
Non-augmented	Pre-trained	2.415000	10	Extended

Table 1: BLEU scores for augmented and non-augmented data using different models

It can be seen that the best performance is obtained with the pretrained model and the augmented data. With that in mind, we decided to furtherly enhance our experiments and see if the results can be improved in different environments, those being: using the custom created tokenizer with the base model and with the removal of the diacritics for the Romani data and using the already existent tokenizer for the pre-trained model with no additional changes and the removal of the diacritics for the Romani data.

The new results can be seen in Table 2

MarianMT Results				
Data	**Model**	**BLEU**	**Epochs**	**Tokenizer**
Augmented and preprocessed	Base	2.2694901	10	Custom
Augmented and preprocessed	Pre-trained	**19.1306000**	10	Non-modified

Table 2: BLEU scores for augmented and preprocessed data using different models

Results		
Model	**BLEU Score**	**Epochs**
NLLB-200	**38.2**	10
Llama-3.2-3B Pretrained	1.6	N/A
Llama-3.2-3B Fine-tuned	21.3	4

Table 3: BLEU scores for Romanian to Romani translation using different models.

Llama-3.2-3B appears to have minimal knowledge of the Romani language, achieving a score of 1.6 without any fine-tuning. The best performance was obtained using the NLLB-200 model, which achieved a BLEU score of 38.2, followed by Llama-3.2-3B with a score of 21.3. MarianMT, despite being a solid sequence-to-sequence model, underperformed compared to the newer models.

3 Limitations

The primary limitation of this work remains the availability of high-quality training data. A larger, well-verified corpus (millions of sentences) would be necessary to further improve performance. While data augmentation techniques helped, the lack of extensive parallel data hindered further progress. Additionally, fine-tuning large models like Llama-3.2-3B required significant computational resources, making experimentation more challenging.

4 Conclusions and Future Work

This work explored various approaches for Romanian to Romani machine translation, testing both traditional Sequence2Sequence models and more advanced multilingual and generative models. The results showed that models explicitly designed for multilingual tasks, such as NLLB-200, performed best. Future work should focus on increasing the dataset size, improving tokenization strategies, and experimenting with more powerful language models to enhance translation quality further.

4.1 References

References

Christos Christodoulopoulos and Mark Steedman. 2014. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49:1–21.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Aji, Nikolay Bogoychev, André Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in c++. pages 116–121.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.