Machine Translation Romanian Dialects

1st Semester of 2022-2023

Claudiu Creanga
ccreanga@s.unibuc.ro

Lucian Istrati
listrati@s.unibuc.ro

Abstract

This is a collection of tools that are helpful in doing translation work between Romanian sub-dialects. It also has several analyses that show the differences between sub-dialects, or as they are called in Romanian: graiuri.

Use the project

1. Download the code from this public repository https://github.com/lucianistrati/Machine-Translation-Romanian-Dialects

2. You can also download models from huggingface: https://huggingface.co/fmi-unibuc

3. To use the "oltenizator" program which changes the tense from passe compose to passe simple, use this command inside src/oltenizator subfolder: python tense_changer.py -s "A aranjat camera."

4. To change it from passe simple to passe compose add the -r flag: python tense_changer.py -s "Aranjai camera." -r

5. SpacY doesn't detect very well the passe simple tense, so the reverse swapping doesn't work as well as the first one.

6. Analysis.py can be used to show the most common used words by every Romanian sub-dialect. It shows also plots with the data.

7. Train_word2vec.py is used to train a word2vec CBOW model over all the books in every dialect;

8. compare_anns.py is used to compare how accurate were the Speech-To-Text transcriptions between Sonix-Ai and Vatis-Tech solutions. This is a place where new contributions can be made as the current analysis lacks in depth.

9. mat.py is used to obtain a similarity matrix between the overlaps of each dialect;

10. Translation.py has all the translation functionality needed to translate from one dialect to another;

11. detect_dialect.py is used to detect the dialect from a text.

12. Utils.py has several dictionaries that contains useful rules for translation as well as mapping of the videos and the books to dialect labels.

13. Train_model.py is used to trains a model to be able to classify in what dialect a text is in.

1. **Data Acquisition**

   Scrap the wikidictionary website for a list of Romanian verbs.

   Scrap the conjugari.ro website for the rules by which a verb can be conjugated: regular or irregular verbs.

   Manually clean these datasets.

   Collect books and texts from each dialect. We created a first dataset of sub-dialects books: RoBoDi.

   Collect audio from speakers of Romanian sub-dialects and start making a first dataset in this field: RAuDI (posted on huggingface).

   Scrap dexonline for words and their dialect (work in progress).

2. **Evaluation**

   Evaluate the systems manually.

   Spot where it makes mistakes and refactor code.

   Due to a lack of data we found it hard to evaluate it automatically.

1 Introduction

Because there are very few analyses at sub-dialect level in Romanian we thought about this project as a collection of tools to provide and enable analyses in this domain.

We are trying to solve 3 problems:

- to make a program that can detect a Romanian sub-dialect from a random Romanian text.
- to make a program that can evaluate how well current speech to text tools work with Romanian sub-dialects.
- to make a program that changes a text from a dialect to another.
- because of lack of data we created a dataset of regionalisms and arhaisms (RoAcReL): 1942 rows and 47 columns: ['Word', 'Meaning', 'First mention', 'IsInDexOrNot', 'County/Region', 'IsItUsedNow', 'Source']. The data was collected from the following regions: 'Dobrogea', 'Muntenia', 'Moldova/Transilvania', 'Transilvania', 'Ardeal', 'Maramureș', 'Bucovina / Republica Moldova', 'Bucovina', 'Moldova', 'Comuna Suharău, Județul Botoșani', 'Comuna Șerbănești, Județul Olt', 'Oltenia', 'Sudul Moldovei', 'Banat';
- we didn't find analyses done on these problems before.

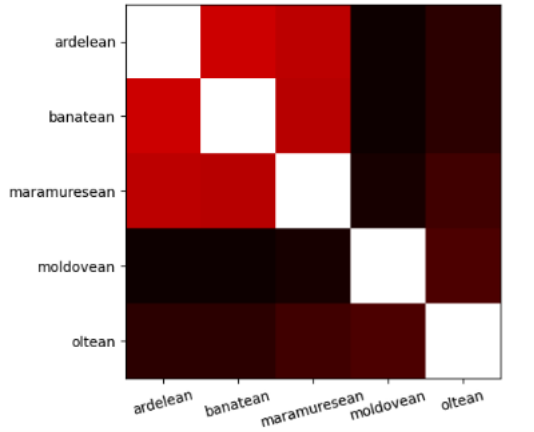
The tense changer gives very good results:

Tu ai plecat in parcare.	Tu plecași in parcare.
Noi am fost la plimbare.	Noi furăm la plimbare.
Ei au negat minciuna.	Ei negară minciuna.
Au negat minciuna.	negară minciuna.
Eu am argumentat bine.	Eu argumentai bine.

We labelled a text or book with a certain dialect and got the following results:

- 101 Basme Romanesti 'ardelean': 0.30, 'banatean': 0.25, 'maramuresean': 0.32, 'moldovean': 0.06, 'oltean': 0.17
- Radu Rosetti, Parintele Zosim 'ardelean': 0.28, 'banatean': 0.24, 'maramuresean': 0.30, 'moldovean': 0.05, 'oltean': 0.12
- Povesti populare romanesti 'ardelean': 0.37, 'banatean': 0.31, 'maramuresean': 0.27, 'moldovean': 0.02, 'oltean': 0.08
- Comorile poporului Radulescu Constantin Bucuresti 1930 'ardelean': 0.31, 'banatean': 0.25, 'maramuresean': 0.25, 'moldovean': 0.02, 'oltean': 0.07

We analized the similarity of the vocabulary of each sub-dialect and found that, as expected, that we can split them in Nordic and Sudic groups:



Lucian Istrati learned how to:

- choose metrics for comparing dialects both verbally and in texts;
- get creative with collecting data for dialectal translation tasks;
- analyse data such that you can outline differences between dialects;
- create a rule based translation system;

Claudiu Creanga learned how to:

- collect data for verb conjugation in Romanian;
- build a tense changer based on rules in Romanian;
- train models for dialect detection;

2 Approach

The code is public and can be used from here: https://github.com/lucianistrati/Machine-Translation-Romanian-Dialects The models are here: https://huggingface.co/fmi-unibuc Tools that we used:

1. SpacY and nltk for different NLP tasks like POS and Morphology tagging.
2. Dexonline, youtube, conjugari.ro for data gathering.
3. project was done in python.

The training was done in Google collab.

3 Limitations

The biggest limitation we have is that we didn't manage to use the full capabilities of dexonline. If we used more data from there we could have improved our model more.

Regarding our Speech-to-Text task, there were only a few tools available in Romanian: Google, Sonix-ai and Vatis-tech. All 3 tools required a paid subscription after the first couple of minutes of free use.

4 Conclusions and Future Work

We liked the project because it provided us the opportunity to do novel research for Romanian language.

We would like to learn how to properly train a machine translation model end-to-end.

Like we said before, the best place to do future contributions in this project would be to use more data from dexonline and improve the model.