Heuristic analysis of the linguistical differences between the Romanian written language in Romania and the Republic of Moldavia

Alexandru Tindeche
alexandru.tindeche@s.unibuc.ro

Florin Dinu
florin-silviu.dinu@s.unibuc.ro

Abstract

The purpose of this project is an objective analysis of the linguistic differences between the Romanian and Moldavian written language. The popular belief is that the language are in effect extremly similar but this paper shows that there are vocabulary and parts of senteces differences that can be successfully used to train a classifier.

1 Introduction

It is clear for every native Romanian speaker that the language spoken in Romania is much different from the one spoken in Moldova. Despite essentially being the same language, historical, political and social factors have influenced the two variants, leading to differences to vocabulary, spelling and possibly syntax. This project aims to conduct an objective analysis of the linguistic differences between Romanian and Moldavian and to see if there are any significant differences in the written language also.

The motivation behind this project is to shed light on the subtle yet significant differences observed in a language used in two geographically and politically distinct regions.

This paper will take a handful of approaches towards finding the differences and similarities between the two written languages. Everything will be based on newspaper articles that have been scraped from Romanian and Moldavian sources which will constitute the corpus.

One such approach is the statistical analysis of the corpus of text in which the text itself is investigated. This analysis only serves as a preamble for the other methods.

Another approach is training a binary classifier that takes as input different kinds of features (i.e. words on their own, parts of speech dependent on context or not). This classifier has the aim of distinguishing between the two written variants and the effectiveness of this classifier can serve as a quantitative measure of the differences between the two languages. If a classifier has a good accuracy, it is clear that the two languages are different enough to be distinguishable, in a significant way.

For distinguishing parts of speech, the paper uses 2 libraries. spaCy (Matthew et al., 2020) provides tokenization, dependency trees and parts of sentence extraction. This will be used in the parts of speech classifier. NLP-Cube (Boroș et al., 2018) is another library from which the same functionalities will be used in order to compare the results.

The paper also analyses the distribution of parts of speech in a sentence and provides statistical comparisons between the two written languages.

In the end the paper will try to find out if 5-grams can be successfully used to generate text in both the written languages. The output will be qualitatively analyzed in order to check for meaning and coherence

2 Approach Summary

First if all the paper analyzes different features of the language, both dependent and independent on context.

One such approach that is not dependent on context is analyzing the mean, median and standard deviation of the articles. The paper does this by "looking" separately at the mean, median and std of the article length, not taking into account the categories, and then the same analysis but across 16 categories (politics, economy, social, sport,

culture, news, entertainment, technology, health, auto, tourism, lifestyle, education, local news, justice, miscellaneous).

Then, we try to train two binary classifiers that are of the same type but are trained on different features. One of them is trained on independent words as features and the other is trained on parts of speech (dependent and independent on context; here we used spaCy and NLP-Cube).

Another approach is analyzing the dependency trees and examine the most frequent grammatical structures from every class.

The project also implements a model that generates sentences, using statistics, trying to imitate the two language variations.

3 Approach

This paper uses both statistical and natural language processing approaches, which will be discussed in the following paragraphs.

3.1 Dataset

The data that has to be analyzed consists of a corpus of articles collected from 10 different publications from each country (Romania and Moldova) - approximately 4632 from Romania and 13120. The data was collected using a Python script that utilizes the Selenium library for web automation and text extraction. For every article, metadata like the publish date, title, category and sometimes the view count was included. The data is saved separately in .txt files and an SQLite database for easier access.

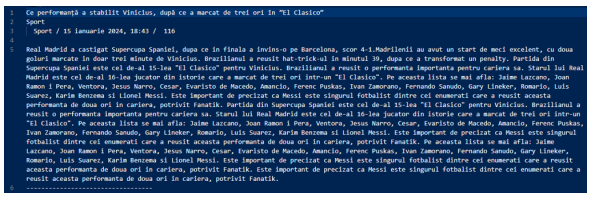

Figure 1: Data collection

3.2 Statistics

As part of a first take on the dataset some statistics were calculated using NumPy (Harris et al., 2020). The mean, median and standard deviation for each country shows no significant differences between the two as seen in figure 2.

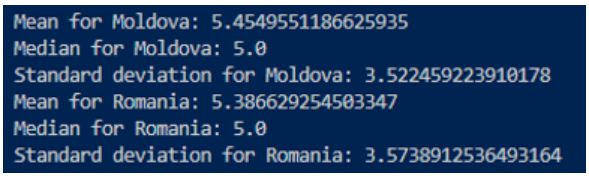

Figure 2: Statistics for the two countries

Moreover, refining the statistical approach to categories no significant differences were found as seen in figure 3.

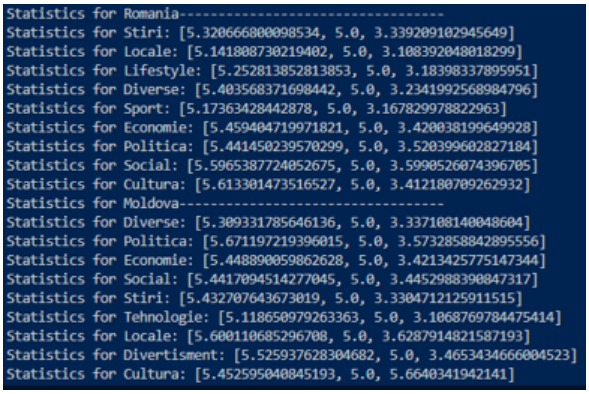

Figure 3: Statistics for the two countries by category

3.3 Binary classifiers

Regarding the binary classifiers mentioned above, there are three classifiers, identical as structure, but trained on different data. The classifier used is Logistic Regression with L2 regularization. The defined pipeline also includes a TfidfVectorizer used for data preprocessing, more specific, to convert a collection of raw documents into a matrix of TF-IDF features (Term Frequency-Inverse Document Frequency).

TF-IDF is a numerical statistic used to reflect how important a word is to a document in a corpus. The parameter "min_df" is used to set the minimum number of documents in which a term must appear to be included in the vocabulary. The parameter "min_df" is used to set the maximum number of documents in which a term must appear to be included in the vocabulary (for example 0.7 means that terms appearing in more than 70% of the documents will be ignored. The argument

stop_words eliminates the words that do not carry much meaning; this can greatly reduce noise. The term clf stands for classifier. In the case of this paper these are the parameters for the vectorizer: $min\_df = 3, max\_df = 0.7, max\_features = 10000, stop\_words = get\_stop\_words('ro')$ (Savand, 2024).

There is also a grid search algorithm to automatize the process of hyper-parameter tuning. Both models are trained on 916 articles, an equal amount, to eliminate any bias that might happen.

The first model is trained on single words, while the other two are trained on POS as features (one dependent on the cotext and the other not). For the classifier trained on parts of speech, the spaCy and NLP-Cube libraries are used to extract the POSs. Moreover, for the second and third classifier, we replace each word with the corresponding POS from evry sentance and feed it into the mode.

```
sss = StratifiedShuffleSplit(n_splits=10, test_size=0.1, random_state=11)
text_clf = Pipeline(steps=[
        ('tfidf', TfidfVectorizer(min_df=3, max_df=0.7, max_features=10000, stop_words=get_stop_words('ro'))),
        ('clf', LogisticRegression(penalty='l2'))
    ], verbose=True)
parameters = {
    'tfidf__ngram_range': [(1, 4)],
    'tfidf__use_idf': (True, False),
    'clf__C': (0.1, 1, 10),
}
gs_clf = GridSearchCV(text_clf, parameters, cv=sss, n_jobs=-1, verbose=1)
```

Figure 4: Binary classifiers

3.4 Dependency trees

Dependency trees can underline the specific type of speech structure used in both written languages. Both corpuses being based on newspaper articles contain a formal standard language with standard structures.

In order to construct the dependency trees, spaCy (Matthew et al., 2020) was used. First step was the sentence division of the text, then all sentences with more than 3 words were selected. This is in order to ensure that the data is not corrupted by inline advertisments or formatted subtitles that were not marked as such inside the article.

Then 2 different approaches were used. The first one is to use only the parts of speech disregarding their place inside the sentence. The second is to keep the place inside the sentence. For both approaches the number of repeated sentences, as in sentences that have the same parts of speech in any order or in that specific order, was counted and stored in a dictionary.

Most common sentences were extracted and their dependency graph plotted. The most common parts of sentence without position can be seen in figures 5 and 6 and those with position in figures 7 and 8.

```
Moldova
ADP ADP ADV AUX NOUN NOUN PRON PUNCT 121
ADP ADP AUX NOUN NOUN NOUN NOUN PRON PUNCT PUNCT 115
ADJ ADP ADP ADP ADP ADP ADV ADV AUX AUX DET DET NOUN NOUN NOUN NOUN NUM PRON PROPN PUNCT PUNCT VERB 114
```

Figure 5: Romania with sentence placing

```
Romana
ADJ ADP ADP ADP ADP AUX DET NOUN NOUN NOUN PRON PRON PUNCT PUNCT PUNCT SPACE VERB VERB VERB 517
ADJ ADP ADP ADV ADV AUX DET NOUN PROPN PUNCT SPACE 166
ADP ADV ADV NOUN NOUN PRON PUNCT PUNCT SPACE VERB 21
```

Figure 6: Moldavia with sentence placing

```
Moldova
NOUN PRON AUX ADP NOUN PUNCT NOUN ADP NOUN PUNCT 114
PRON AUX ADV DET ADV ADJ NOUN AUX VERB ADP PROPN PUNCT ADP DET NOUN ADP NUM ADP NOUN ADP NOUN PUNCT 114
PRON ADV AUX ADP NOUN ADP NOUN PUNCT 113
```

Figure 7: Romania without sentence placing

```
Romana
NOUN ADP NOUN ADP PRON PRON AUX VERB ADP NOUN ADJ PUNCT VERB ADP PUNCT SPACE DET VERB PUNCT 517
ADV AUX DET NOUN ADV ADP ADJ ADP PROPN PUNCT SPACE 166
NOUN PRON VERB ADV ADV PUNCT ADP NOUN PUNCT SPACE 21
```

Figure 8: Moldavia without sentence placing

The common sentence structure is only present in a small percentage of the entire corpus which is expected. This is seen for the place inside the sentence agnostic approach in figure 9 and for the other in figure 10.

```
Ro-Md common: 714
Ro common percentage: 3.58%
Md common percentage: 1.73%
```

Figure 9: Common with sentence placing

None of the most common parts of speech distributions in each language were found in the other as shown in

3.5 Text generator

Text generation was used by using 5-grams on the whole Romanian and Moldavian corpus and

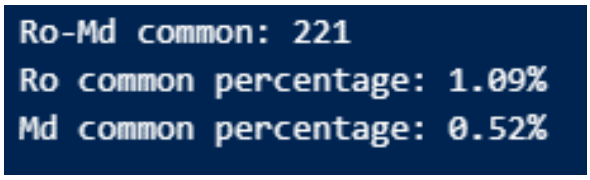

Figure 10: Common without sentence placing

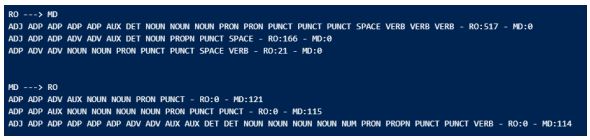

Figure 11: Most common with sentence placing

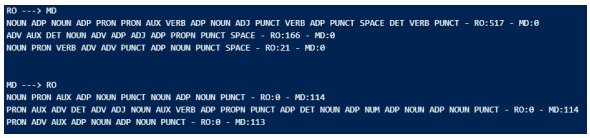

Figure 12: Most common without sentence placing

counting the frequency of the 5th word in sequence to the first 4. Because of spaCy's (Matthew et al., 2020) limitations, both corpuses had to be trimmed at 1.000.000 characters.

The first 4-gram is chosen at random from the 4-grams that start with uppercase, quote marks or opened parenthesis and contain no intermediate or final punctuation mark. Then the most frequent word is chosen and the 4-gram becomes the last 4 words of the newly formed text. It continues until a final punctuation mark is reached or there are no more candidates for the next word (a full stop is issued). The generated texts can be seen in the figure

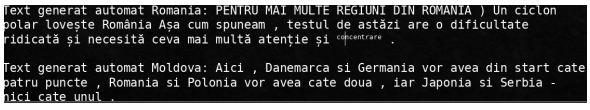

Figure 13: Generated sentences

Overall, the project was developed over the course of two weeks of constant work, with frequent meetings between the team members and checkups with the project coordinator.

4 Conclusions

According to the results presented in the Approach section, there is a clear difference between the Romanian written language in Romania and Moldavia both at the vocabulary level and at the parts of speech level. This difference can be successfully used to train classifiers in order to ascertain the nationality of the author.

This paper uses a small but significant dataset for comparison. In future papers a larger dataset can be used both for comparisons and for text generation. The text generation might be lacking both due to the limitation of the model and the limitations of the dataset therefore a larger dataset may pinpoint the problem or alleviate it.

Another future improvement can be the use of a state of the art model for language analysis. This approach will make better judgments and will be able to produce better classifiers.

In order to improve the text generation task a better model, like GPT-Neo (Black et al., 2021) can be used.

For future work, a corpus of informal speech can be used since in theory informal speech varies vastly from the written language. In this aspect an accent based approach can be used to distinguish between two similar accents, the Moldavian accent from Romania and that from the Republic of Moldavia.

In the end, the paper proves there are significant differences between the two written languages and a classification model can be trained contradicting the popular belief of two very similar languages.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Tiberiu Boroș, Stefan Daniel Dumitrescu, and Ruxandra Burtica. 2018. NLP-cube: End-to-end raw text processing with neural networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 171–179, Brussels, Belgium. Association for Computational Linguistics.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg,

Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Honnibal Matthew, Montani Ines, Van Landeghem Sofie, and Boyd Adriane. 2020. spacy: Industrial-strength natural language processing in python.

Alireza Savand. 2024. Python Stop Words.