Faculty of Science and Engineering

Information Retrieval and Text Mining

2223-KEN4153

Unraveling Entities, Interactions and Emotions in Fyodor Dostoyevsky's 'The Idiot'

Project Report

31/05/2023

Ismael Gomez Garrido i6334270

Abstract

This report is based on an information extraction process over the Dostoyevsky's novel 'The Idiot', Before any information extraction it was necessary to preprocess the data. This preprocessing included tasks like the separation of the book into chapters, tokenization, Named Entity Recognition (NER) and entity unification. After that, three different visualizations were made with the intention of extracting valuable information: counting of character appearances (pure count and tf-idf score), estimation of the characters' interactions with each other and text classification into the main 7 emotions (6 from Ekman + neutral). For the text classification 3 pretrained BERT based models were used and its performance was calculated by the accuracy on a random subset of paragraphs manually annotated. After analyzing the results, the visualizations helped to understand the characters importance in the book along the chapters. However, further research is possible to achieve better results.

Key words: Text Mining, Information Retrieval, NER, text classification, BERT

Table of contents

1. Introduction

This project aims to extract valuable information from Fyodor Dostoevsky's book "The Idiot" using techniques from Information Retrieval and Text Mining. The project focuses on data preprocessing and information extraction using visualization techniques.

Through data preprocessing, the raw text is transformed into a structured format for effective analysis. This involves tasks like removing punctuation, tokenizing the text, and normalizing it. The project then extracts meaningful information, such as named entities and main characters, to gain insights into their roles and relationships.

Sentiment analysis is another important aspect of the project. By analyzing the predominant sentiments in the text it's possible to have a deeper understanding of the book's mood and the sentiments towards different characters and events.

On the one hand, the project employs visualization techniques to enhance the interpretability of the extracted information. On the other hand, quantitative are also used to have an objective to evaluate the information and the knowledge extracted from 'The Idiot'.

2. Dataset

The Idiot is a novel written by Fyodor Dostoyevsky in 1868-1869. "The Idiot" explores deeply some of the most important human sentiments such as love, loyalty, innocence and naivety, mental illness, and the complexities of human relationships. It explores them using as a frame the Russian high society from the latest 18th century.

The book contains around 240 000 words, so it constitutes a solid source from where relevant information can be extracted. In the case that the amount of text is not enough a possibility is to use more books by Dostoyevsky, all of them treat some deeply aspects of the human mind so a general knowledge can be extracted from them as well.

3. Preprocessing

The first part of the project was to download the selected dataset. The book is available in the Project Gutenberg's library [1] so it was not a problem to extract the raw data in a .txt format.

Before starting any text mining operation like tokenization or named entity recognition, a process of cleaning and organizing was necessary:

- The beginning and the end of the book contained irrelevant information about the author, the year of publication and the licenses associated to Project Gutenberg. This information was removed manually.
- The characters belonging to the jump of line "\n" had to be replaced to have a flat string. However, the double jump of line present between paragraphs allowed to separate the text in paragraphs. This division was used later for different purposes.
- The structure of the book consists of 4 parts, each of which starts like PART I., PART II and so on. In turn, each one of the parts is composed of a different number of chapters, which are numbered with roman numbers. The division in chapters presented some difficulties extracting the exact string of every chapter but with the use of a regular expression pattern it

was possible to perform the division successfully. A list with a total of 50 chapters was obtained.

- Part I → 16 chapters
- Part II → 12 chapters
- Part III → 10 chapters
- Part IV → 12 chapters

3.1. Named Entity Recognition

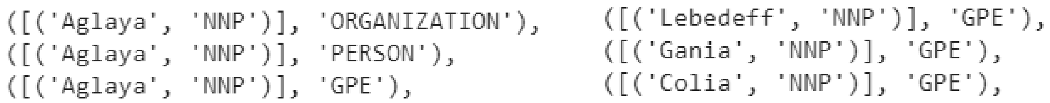
Named Entity Recognition (NER) is a subtask of natural language processing (NLP) that aims to identify and classify named entities in text into predefined categories such as person names, organization names, locations, dates, etc. In this project this was useful for entity extraction. It allowed to unify some persons or locations so then it was possible to extract valuable information from them.

Word tokenization was the first technique to be used in this process. In order to preserve the maximum of details of the information neither stop words, lemmatization or stemming were applied.

A technique used to extract the entities was chunking, it consists in grouping small pieces of information. Chunking uses Part of Speech (POS) tagging and the regular expressions to extract the chunks.

The main resource used for this task was the NLTK library (Natural Language Toolkit), with the pretrained name entity chunker *ne_chunk()* it was possible to extract a list of all the entities present in the book. This list presented some problems that were treated as follows:

- **Duplicates:** The obtained list presented a lot of duplicated entities. This was expected because the chunker was fed sentence by sentence (a process of sentence tokenization was applied), so the main characters appear in many sentences.
- **Wrong entity categorization:** the NLTK chunker is able to provide a label to classify the type of entity recognized: Person, Geopolitical Organization, Location, Facility … These labels were mostly wrong, and, in many cases, the same entity was categorized as different kinds of entities, Figure 1. This error can be caused by the lack of context, the chunker is fed only sentence by sentence. Because of this the use of the entity categorization labels was discarded.

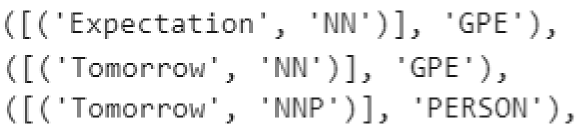


```
([('Aglaya', 'NNP')], 'ORGANIZATION'),
([('Aglaya', 'NNP')], 'PERSON'),
([('Aglaya', 'NNP')], 'GPE'),
```

```
([('Lebedeff', 'NNP')], 'GPE'),
([('Gania', 'NNP')], 'GPE'),
([('Colia', 'NNP')], 'GPE'),
```

Figure 1. Examples of wrong entity categorization.

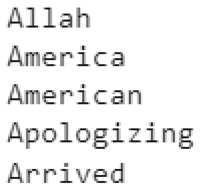
- **English words:** Even after the removal of the duplicates and the categorization labels a lot of the entities recognized were not real entities. To remove these entities the English words corpus from NLTK was used, this corpus contains more than 230 000 words from the English language. If a recognized entity was in that corpus it was deleted from the list. Some of the removed words with this method are shown below in Figure 2.



```
([('Expectation', 'NN')], 'GPE'),
([('Tomorrow', 'NN')], 'GPE'),
([('Tomorrow', 'NNP')], 'PERSON'),
```

Figure 2. Entities removed after using English words corpus.

- **Synonyms:** After analyzing the resulting entities, some of them weren't real ones, like conjugation of some verbs or the plural of some words. To solve this problem, I used the synset structure from NLTK, it's connected to WordNet, and it returns synonymous words. An entity was removed from the list if it had some synonym. A side effect of this technique is that some real locations were removed from the entity list, but the location extraction wasn't a priority in his project. In figure 3 some of the filtered words are shown.



```
Allah
America
American
Apologizing
Arrived
```

Figure 3. Entities removed using synsets.

- Finally, reviewing the list of removed entities, 2 entities were added back because they corresponded to the name of 2 main characters of the book.

In the next table the evolution of the NER process is shown. A total of 391 different entities were recognized at the end of the process, but 2 things have to be considered. The first one is that, as will be exposed later, some characters are referred to with different names, so they have duplicated entities. Second one is that a lot of the unique entities have almost no presence in the book, only 25 have more than 100 appearances and less than 100 entities have more than 10 appearances in the whole book.

State of the process	Nº of entities
Original list	6409
Removal of duplicates and labels	776
Filtering by English words corpus	483
Filtering by synsets	389
Manually added	391

Table 1. Evolution of the NER process.

3.2. Entity unification

The objective of the NER was to identify the unique entities so that valuable information can be extracted from the text. Therefore, after identifying them, it was necessary to unify them under the same name. Many characters are called by different names along the book, in the table below the variations of some names are exposed:

Character	Name variations
Prince Muishkin	"Prince Lef Nicolaievitch", "Prince Lef", "Lef Nicolaievitch", "Lef", "Mr. Muishkin"
Aglaya Ivanovna	"Aglaya Ivanovna Epanchin", "Aglaya Ivanovitch", "Aglaya Epanchin", "Aglaya"
Nastasia Philipovna	"Nastasia Philipovna Barashkoff", "Nastasia"
Rogojin	"Mr. Rogojin", "Parfen Rogojin", "Parfen Semeonovitch", "Parfen Semionovitch", "Parfen."
"Lebedeff"	"Lebedef", "Mr. Lebedeff", "Lukian Timofeyovitch", "Lukian"
"General Epanchin"	"Ivan Fedorovitch Epanchin", "General Ivan Fedorovitch", "Ivan Fedorovitch", "Fedorovitch Epanchin", "Fedorovitch", "Epanchin"
Lizabetha Prokofievna	"Lizabetha", "Elizabetha", "Elizabetha Prokofievna"

Table 2. Variations of the names of some characters.

A total of 16 characters were chosen to perform the entity unification, the rest of characters weren't considered relevant enough due to its low number of appearances.

To perform the substitution in the book several regular expressions were used, there were 2 main problems encountered:

- **The order of substitution**: It was important in which order the entities were going to be replaced for the original entity name. This happened, for example in the case of the Prince Muishkin: if the first replacement is by looking at "Prince Lef", then "Prince Lef Nicolaievitch" will be substituted by "Prince Muishkin Nicolaievitch".
- **Exceptions**: In some cases, a character is named only by its family name, but there are other members of the family in the book. In order to avoid ambiguity, regular expression patterns were used. For example, in the case of the General Epanchin: the general has 3 daughters and his wife. So, the regular expression captures Epanchin only when is not preceded by either the name of one of the daughters or Mrs. or Madame.

4. Information retrieval

4.1. Characters appearances

Once each entity is unified under a unique name is possible to make visualizations to understand the importance of the characters along the book. To show this importance I calculated two metrics for every main character on each chapter:

- **Counts:** The number of times that the character appears in the chapter
- **TD-IDF score:** Used to measure the importance of a term on a corpus of documents. It's a balance between the term frequency and the inverse document frequency, this is the formula:

$$w_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log_{10}(\frac{N}{\mathrm{df}_t})$$

In the figure below (figure 4) the raw appearances of every main character are shown at every chapter, the black dotted lines represent the division between the four parts. It's easy to determine that the main character is the Prince Muishkin, the title of the book is in reference to him. The appearances of the two lovers of the Prince (Nastasia Philipovna and Aglaya Ivanovna) are also notable. Besides, it is possible to see that Nastasia appears principally in the first part and in the end, in the middle parts appears intermediately. On the other hand, the importance of Aglaya begins to grow in the third part, which is where the Prince sends letter to her.

Another remarkable observation is that in some chapters the number of character appearances is extremely low, for example in the number 6. The reason for that is that in this short chapter the Prince is telling a history about a girl he met when he was in treatment in Switzerland. This girl is not relevant to the main argument of the book and the Prince just tells it to illustrate his experience in the country.

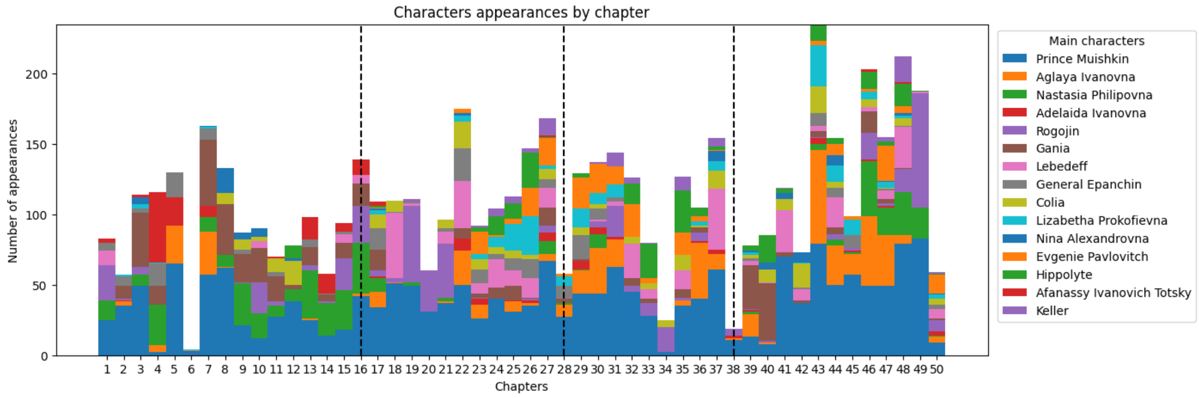


Figure 4. Character appearances by chapter.

In figure 5, referring to the tf-idf score the distribution clearly changes with respect to the previous graphs. The Prince Muishkin doesn't appear in the graph, this is because he's mentioned in every chapter so his inverse document frequency is 0. However, other characters with small importance before show now a bigger influence. This is the case for Lizabetha Prokofievna.

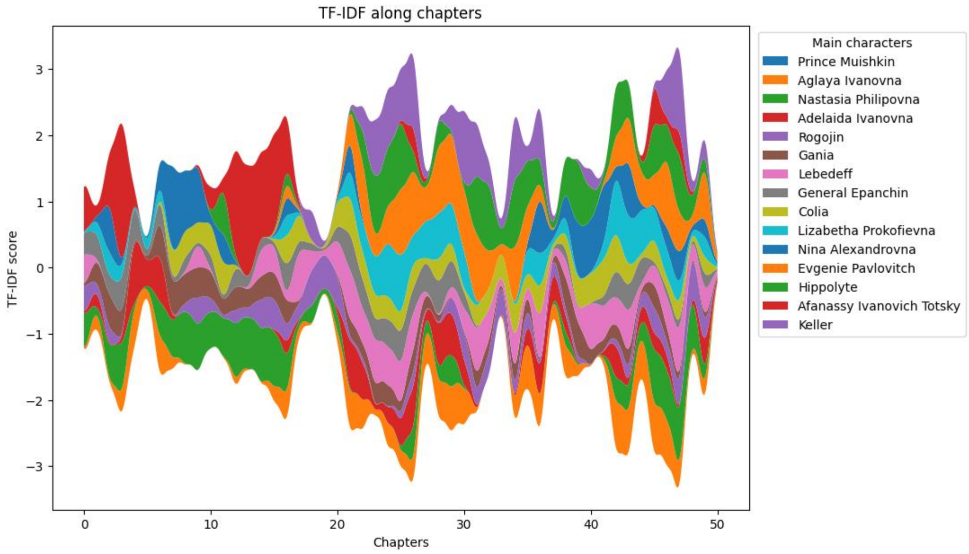


Figure 5. tf-idf scores along chapters.

4.2. Interactions between characters

Another piece of information that can illustrate the relationship between the characters is the number of interactions that they have between each other. It is practically impossible to measure in a precise way when two characters are interacting between each other, but with some approximate rules is possible to have an estimation.

In this case te rule to determine if 2 characters were having an interaction was to check if their names are mentioned within a given number of words. This window size parameter varied between 100 and 1000 tokens, but the differences in the results were very small.

This basic rule has some evident problems and there are different cases where it'll result in a false positive (when a name of a character is mentioned in a conversation, or the narrator just mention both names…). Despite that, I think that reflects in a very approximate way the interactions between characters. If one character mentions another in a conversation, it means that they do have some kind of relationship so it can be considered some kind of interaction.

One of the difficulties found during the implementation was to calculate the token distance and to identify the tokens of those entities with composed names (Prince Muishkin, Aglaya Ivanonva…). The solution was to merge those tokens: Prince_Muishkin, Aglaya_Ivanovna…

The figure below is a Chord diagram, and it reflects the number of interactions counted between the characters. In this example the window size is 1000 tokens but the only difference with the other window sizes proven is almost nonexistent. For the implementation of this diagram the Flourish studio environment was used [2].

Again, we can see that the main character and the center of almost all the interactions is the Prince Muishkin. Without exception, the biggest interaction of every character is with him.

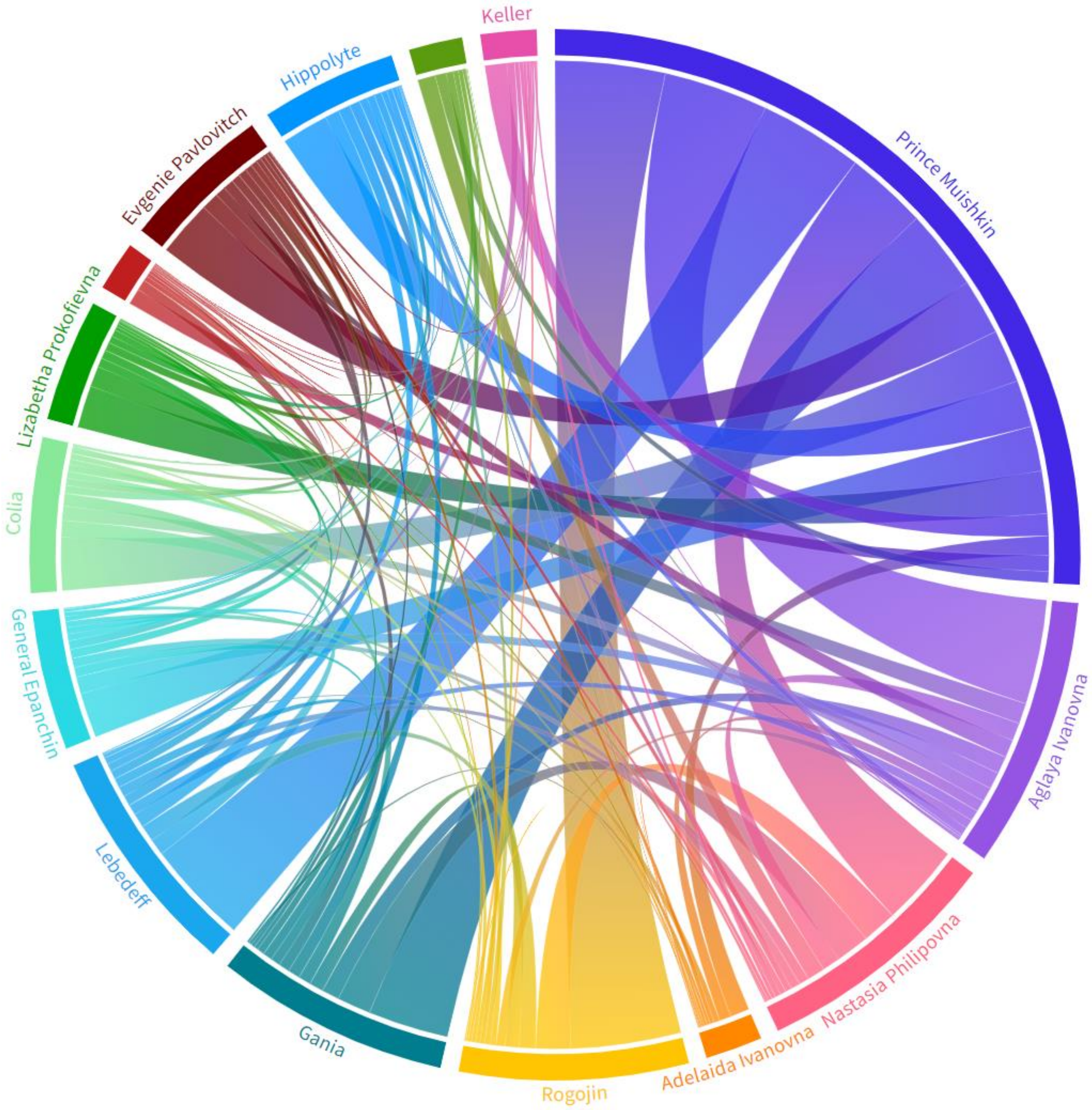


Figure 6. Chord diagram representing the interactions between the characters.

4.3. Text classification with BERT

The third and last analysis of this project was the text classification with BERT-based models. In the Hugging Face platform, I found three models for the purpose of text classification whose output labels are the same. These models are:

- j-hartmann/emotion-english-distilroberta-base → Fine-tuned checkpoint of DistilRoBERTa-base [3]
- j-hartmann/emotion-english-roberta-large → Fine-tuned checkpoint of RoBERTa-large[4]
- michellejieli/emotion_text_classifier → Fine-tuned version of Emotion English DistilRoBERTa-base and DistilRoBERTa-base [5].

All three models are fine-tuned versions of DistilRoBERTa and they have been trained with similar datasets. The datasets represent a diverse collection of text types. Specifically, they contain emotion labels for texts from Twitter, Reddit, student self-reports, and utterances from TV dialogues.

Possible emotion labels	
1	anger 😡
2	disgust 🤢
3	fear 😨
4	joy 😀
5	neutral 😐
6	sadness 😭
7	surprise 😲

Text classification

The text classification was performed by paragraphs. To see if there was an evolution of the general sentiment of the book, a random selection of 20 paragraphs from every chapter were analyzed by each one of the three models. The number 20 was chosen because the chapter with less paragraphs in the book is the number 50 with only 21.

The maximum number of tokens that the models accepted was 512. Nonetheless, the paragraphs were selected with a maximum length of 250 tokens to facilitate the classification task. Here are the results:

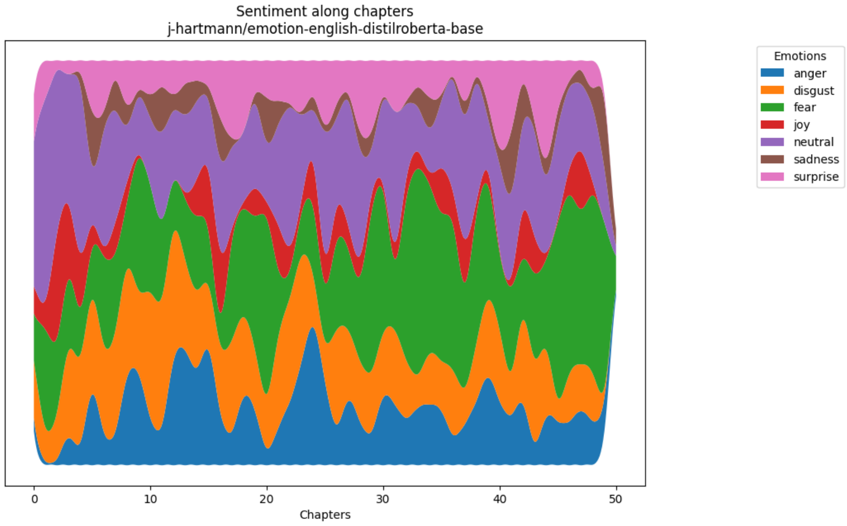


Figure 7. Emotions detected with model number 1.

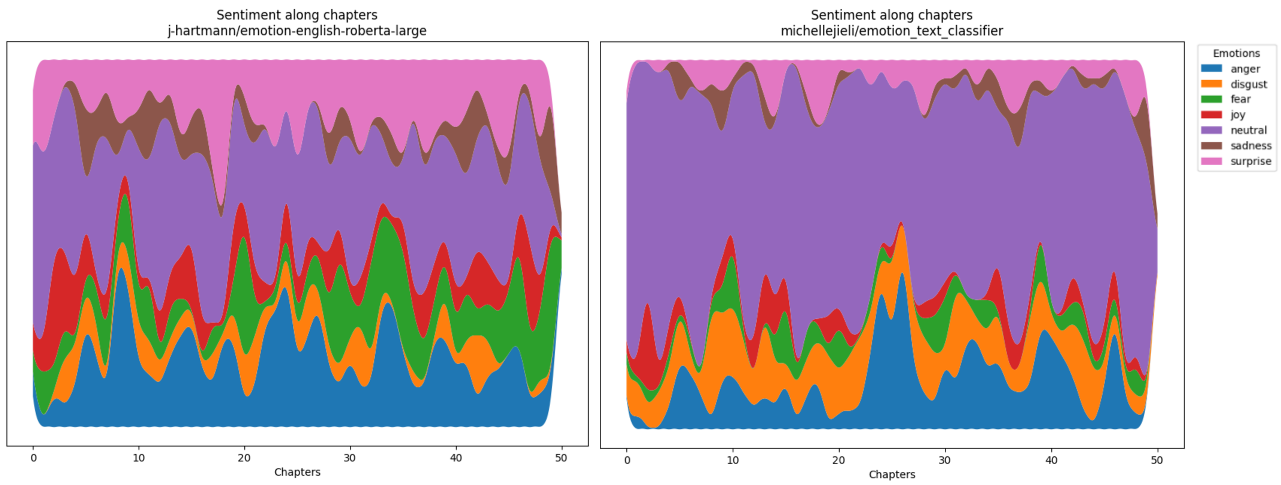


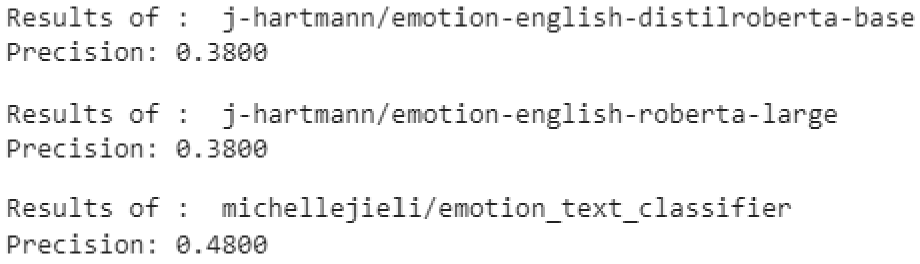
Figure 8. Emotions detected by the model 2 (left) and 3 (right).

After analyzing the results few conclusions were extracted:

- The distribution of the emotions along the chapters is, in general, very different in every model. There is maybe a peak of anger (blue) between chapters 20 and 30 that is common in all the plots. This could be explained by some intense events that take place in that part of the storyline.
- In the first and second plot it seems to be a tendency of fear emotion (green) to grow at the same time the history advances. Although this tendency is not followed by the third model, there are some subplots that in the last parts of the book take on a more serious character and therefore, the feeling of fear could be a little bit more common.
- The number of neutral labels grows from plot 1 to plot 2, but in the plot corresponding to the third model this label is clearly the most common in every chapter. Depending on the training process some models can interpret the same inputs differently, even if they are very similar as is the case here.

Numerical methods

In order to measure in an objective way how accurate are the models used, a random subset of 50 paragraphs (1 from every chapter) was analyzed and manually labelled. The precision score was calculated by the ratio between the labels predicted correctly and the total number of labels. These are the scores that the models obtained:

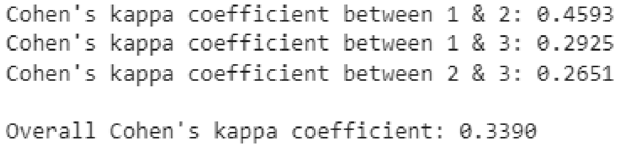


```
Results of :  j-hartmann/emotion-english-distilroberta-base
Precision: 0.3800

Results of :  j-hartmann/emotion-english-roberta-large
Precision: 0.3800

Results of :  michellejieli/emotion_text_classifier
Precision: 0.4800
```

Also, since three models made predictions over the same labels, it was possible to calculate the Cohen's Kappa coefficients between models and the overall one:



```
Cohen's kappa coefficient between 1 & 2: 0.4593
Cohen's kappa coefficient between 1 & 3: 0.2925
Cohen's kappa coefficient between 2 & 3: 0.2651

Overall Cohen's kappa coefficient: 0.3390
```

Analysis of the numerical results

The predictions of the models weren't very successful. Furthermore, the Cohen's Kappa coefficient indicated that the predictions of the models didn't match very often. Two main reasons can explain these results:

- First of all, the datasets with which the models used have been trained don't contain the same kind of text as in this book. This book contains large and complex sentences which sometimes are hard to understand even for humans.
- Second, the way of evaluating the models was by taken the emotion predicted with more probability. I even found problems when labelling the paragraphs manually to select only one emotion to describe the whole sentiment of the paragraph.

Here are some paragraphs from the random subset used to calculate the precision of the models:

Paragraph	My label	Label 1	Label 2	Label 3
"Wonderful!" said Gania. "And he knows it too," he added, with a sarcastic smile.	3 (joy)	3 (joy)	0 (anger)	3 (joy)
"Surely there must be someone among all of you here who will turn this shameless creature out of the room?" cried Varia, suddenly. She was shaking and trembling with rage.	0 (anger)	2 (fear)	2 (fear)	0 (anger)
"I don't want any dinner, thanks, Colia. I had too good a lunch at General Epanchin's."	3 (joy)	1 (disgust)	5 (sadness)	4 (neutral)
"I don't love you, Lef Nicolaievitch, and, therefore, what would be the use of my coming to see you? You are just like a child—you want a plaything, and it must be taken out and given you—and then you don't know how to work it. You are simply repeating all you said in your letter, and what's the use? Of course I believe every word you say, and I know perfectly well that you neither did or ever can deceive me in any way, and yet, I don't love you. You write that you've forgotten everything, and only remember your brother Parfen, with whom you exchanged crosses, and that you don't remember anything about the Rogojin who aimed a knife at your throat. What do you know about my feelings, eh?" (Rogojin laughed disagreeably.) "Here you are holding out your brotherly forgiveness to me for a thing that I have perhaps never repented of in the slightest degree. I did not think of it again all that evening; all my thoughts were centred on something else—"	5 (sadness)	0 (anger)	0 (anger)	0 (anger)
So spoke the good lady, almost angrily, as she took leave of Evgenie Pavlovitch.	0 (anger)	0 (anger)	0 (anger)	4 (neutral)
"Oh, Mr. Lebedeff, I am told you lecture on the Apocalypse. Is it true?" asked Aglaya.	6 (surprise)	4 (neutral)	6 (surprise)	4 (neutral)
"Are you happy—are you happy?" she asked. "Say this one word. Are you happy now? Today, this moment? Have you just been with her? What did she say?"	4 (neutral)	3 (joy)	3 (joy)	0 (anger)

5. Failed experiments

5.1. Coreference and pronoun resolution

The problem of coreference and pronoun resolution was addressed using different techniques. The main one was by using the library fastcoref. Within this library two models were tested: FCoref and LingMessCoref.

With the first one I was able to perform the pronoun resolution and substitute in the text all the pronouns with the supposed correct nouns. But analyzing the results I detected a lot of mistakes of coreference. One error that I saw it was very often that in the detected clusters no noun was detected so everything ended up substituted with pronouns like him, mine, her…

Regarding the second model I obtained good results with small parts of text but when the text length was too long the model returned error. In this book there are a lot of pronouns used along big parts of text and dialogues, so using the model only with few sentences at a time didn't make a lot of difference.

5.2. Topic Modelling

I tried to extract the main topics of the book using LDA and BERTopic but from the returned outputs wasn't possible to extract any valuable information. I think topic modelling doesn't apply very well to the book analysis because, although there are several main topics along the novel (love, loyalty, society…), these ones are too abstract for the models to recognize.

6. Conclusion

General considerations

After the preprocessing tasks, the Named Entity Recognition resulted in the unification of the main entities of the novel. The importance of the main characters, especially of Prince Muishkin, was visualized and validated by the number of appearances and the number of interactions between the characters.

The performance of the pretrained BERT models wasn't successful in the text classification task, but the models were able to correctly identify the main sentiment of some paragraphs.

Posible further improvements

There exist several ways to continue this project and look for better results:

- With further research on the coreference problem would be possible to obtain more accurate information about the characters' appearances.
- The design of more accurate and specific interaction rules would allow for a better understanding of the relationships between characters.
- Find pretrained models which have been trained with more similar data.
- Consider the top 3 emotions detected instead of just the first one, this will increase the precision score.

7. References

[1] *The Idiot by Fyodor Dostoyevsky - Project Gutenberg*. Accessed: May 31, 2023. [Online]. Available: https://gutenberg.org/ebooks/2638

[2] “Flourish Studio.” https://flourish.studio/ (accessed May 31, 2023).

[3] “j-hartmann/emotion-english-distilroberta-base.” Accessed: May 31, 2023. [Online]. Available: https://huggingface.co/j-hartmann/emotion-english-distilroberta-base

[4] “j-hartmann/emotion-english-roberta-large.” Accessed: May 31, 2023. [Online]. Available: https://huggingface.co/j-hartmann/emotion-english-roberta-large?text=This+movie+always+makes+me+cry..

[5] “michellejieli/emotion_text_classifier.” Accessed: May 31, 2023. [Online]. Available: https://huggingface.co/michellejieli/emotion_text_classifier