Information Retrieval and Text Mining

Group Assignment

Eleftherios Tetteris i6295677

Athanasopoulos Nikolaos i6310104

30/05/2023

Table of contents

Introduction

This project aims to extract interesting and relevant information from Agatha Christie's murder mystery books, using various information retrieval and text mining operations and methods. The main focus for each step will be to compare results between each method, and see the differences between them and how well they worked. The dataset that was used for this assignment was 6 different books (in utf-8 format) from Agatha Christie, namely "Mysterious Affairs at Styles" (1920), "Murder on the Links" (1923), "Plymouth Express" (1923), "Poirot Investigates" (1924), "The murder of Roger Ackroyd" (1926) and "The big four" (1927). Additionally, the project focuses on extracting valuable insights from the texts, enabling the team to unfold the mysteries between characters and shed light on the underlying motives,clues and suspense that made Agatha Christie's murder mysteries so captivating. Through the analysis, the aim is to provide a summarised overview of key aspects and important details of those 6 murder mystery novels. By leveraging information retrieval and text mining techniques a comprehensive overview is presented that captures the main essence of the stories, allowing users to gain a quick understanding of the narratives without delving into the complete books. The complete code, along with every visualization presented here, is available on Github:
https://github.com/LuckySe7enz/IRTM_Agatha_Christie
Table 0 contains the work split between the team.

Eleftherios Tetteris	**Nikolaos Athanasopoulos**
Text Preprocessing	NER with flair
NER with Spacy	Topic modelling
Sentiment analysis	Character tracking

Table 0: Work split between the team members

Motivation

The motivation behind this project lies in the mysterious and intriguing development of the plot. As hinted by the captivating title "Myrder Mysteries" it sparked the curiosity of team, pushing not only to unmask potential suspects but also unfold hidden patterns resulting in second thoughts and misgivings. The book exploration will be covered not just through traditional methods but diving into capabilities of information retrieval and text mining analysis.

Through the application of various methods taught during lectures and labs a thorough clean of the texts as well as extraction of various entities such as names, locations, dates and times from texts is performed, allowing the team to establish connections between characters using various predicates such as emotions and sentiments.

Considering the unique nature of murder mysteries, another focus is on extracting specific relations such as identifying the suspect characters for the murders or acts of violence overtime as described throughout the storyline. Additionally, the team explores the spatial and temporal dynamics of the characters which aims to analyse how the characters move and interact within the story's physical locations and how their actions unfold over time. By doing this, the aim is to understand the relationships between characters, their movements across different places as well as the progression of the events through the books.

Another crucial aspect of the investigation is topic modelling. The aim is to uncover and track the different topics present through the books. This exploration will provide valuable insights into the underlying themes and motifs that arise throughout the story.

To assess the effectiveness of the models and techniques used, a comparison of the results with manual annotations made by the team. Classification metrics are utilised to quantify the performance of the models in tasks such as named entity recognition, sentiment and emotion analysis. Although lacking an annotated corpus based on the nature of the books, the recognition of the benefits of evaluating the performance of the model in such a way cannot be diminished .

In summary, this project embarks on a comprehensive exploration of 6 murder mysteries, by leveraging information retrieval and text mining techniques to explore hidden patterns, analyze character dynamics and identify potential suspects. By engaging with the vast audience of readers who have enjoyed Agatha Christie's books, the investigations aim to offer a profound comprehension and captivating insights into the enigmatic narratives of Agatha Christie. The goal is to provide a richer understanding of her storytelling, catering to the curiosity of the diverse range of readers as well as fueling the curiosity of non-readers.

Methods and Approach

The main steps for this assignment were:

Text extraction, Pre-processing of text, Entity extraction-recognition, Topic Modelling, Co-reference handling, Sentiment Analysis of characters, Tracking of certain characters through the books, Quantitative results for every step and Visualizations.Each step will be discussed in detail in the next sections.

Text Extraction

All of the texts are available from the Project Gutenberg site (https://www.gutenberg.org/) in .txt format with a UTF-8 encoding and they feature the same main character, Hercules Poirot, which is one of the key focus points of this assignment. The files were downloaded and opened in a text editor for inspection. From this, some very interesting points were highlighted: Each book has an introduction section which introduces the reader to the book

and gives him some information about the language and the publication details. Each book also has an ending section which discusses some legal information, along with some donation links etc. These sections are of course irrelevant to this project, so they had to be deleted. One minor issue presented here was that these sections are not standardized across books, so each one had to be inspected by the team and deleted manually, using text partitioning.

Another main point that needs to be highlighted here is that the text is not annotated, meaning that any model that will be used in further analysis has to be pre-trained. This also means that the results in classification tasks will be analyzed by comparing the results of each model with the manual annotations from the team (as ground truth), on a subset of the data each time. This, of course, means that the performance metrics presented may not be completely indicative of the real performance of the models on this dataset, but since a lot of data points were used in each evaluation, it is believed that the metrics presented in each section should be relatively close to the true ones.

Text Pre-processing

The main issue resolved in this stage was the use of special characters, for example the string “****” between chapters. All these were deleted from the text, since, again, they are not relevant to the topic and, also, they cause problems in the later stages of this assignment if not removed. A total of 15 different special characters were deleted from each text, in order to make it ready for further processing. With this, the text is ready for the analysis which will be presented in the next sections.

Named Entity Recognition

Named Entity Recognition (NER) is a vital part of this assignment, as it allows the users to not only identify when and where the story unfolds but also what characters are involved. Thus, they can focus on specific characters or timeframes and also identify common characters and locations between books. This is especially important for tracking characters through the story, which is another key aspect of this assignment.

Two distinct methods were used for NER: FlairNLP, and Spacy. It should be noted that Spacy was used just as a baseline in order to get a quick and dirty result which can then be compared to FlairNLP. Flair is an open source library from Akbik et al.[1] (2018), which uses contextual string embeddings to extract named entities from the text. Flair provides not only the tagger class for NER, but also a sentence detection class which splits the text into sentences. It achieves an F1 score of 93,06% on CoNLL-03, and it predicts 4 tags: Person, Location, Organization and Misc (other).
Spacy predicts more than these 4 entity types, but they are not so relevant to this assignment, so the focus here is also on these 4 tags. Here, the tags GPE and LOC are merged into 1, since London could either be a Location and a Geopolitical Entity.

For the quantitative results of this section, a set of 24 sentences from all the 6 different books were picked by the team, and both models were tested for comparison. The results can be seen in Table1.

Entity Type	Flair precision	Flair recall	Flair F1	Spacy precision	Spacy recall	Spacy F1
Person	1	0.79	0.88	0.67	0.33	0.44
Location	0.88	0.93	0.90	1	0.33	0.50
Organization	0.33	1	0.50	-	-	-
Misc	-	-	-	-	-	-

Table 1: Named Entity Recognition Results for FlairNLP and Spacy

The Cohen Kappa score for the 2 annotators is 0.88, which indicates that the annotators indeed agree well.

From these results, many interesting insights can be derived. First of all, it is obvious from all the metrics that FlairNLP performs better than Spacy in all tags, probably because of the context captured by the embeddings used. Spacy tends to miss entities altogether, which is a big issue when the main focus points are based on a well-performed Named Entity Recognition. The F1 score for Location and Person, which are the 2 most essential tags in this assignment, are very high for Flair, which is the main reason why this model was used in the sentiment analysis and the person tracking afterwards.

Topic Modelling-Topic River

Topic modelling helps uncover the underlying structure and themes in the text, making it easier to organise and explore the information contained within. It consists of finding lists of words which occur together in the same context and best describe these themes in the books. It is also the key part of producing a topic river, which is a visualization of the themes-topics with the text in chronological order. This topic river helps users easily visualize how strong these topics are from book to book, and in which book they are introduced (or terminated). Before producing these topics, it is the intuition of the team that some topics will be associated with crime, others with suspects and others with the main characters themselves, since these are overall themes that run through murder mystery novels in general.

Again, 2 different algorithms were used for topic modelling: Non-negative Matrix Factorization (NMF) and Latent Dirichlet Allocation (LDA). Both of these need as input the number of topics and some form of representation of the texts as words.This representation was chosen to be Term Frequency-Inverse Document Frequency (TF-IDF), as it is able to capture the important terms thanks to measuring both how frequent a term is in a single document, and how often the term is used in the corpus. The number of topics was decided through experimentation with the number of top words per topic and the number of topics, by

looking at the resulting coherency score, and picking the set that leads to the higher coherency.

LDA works by assuming there are two distributions: a distribution of words on every topic and a distribution of topics per document. Then the algorithm tries to find these distributions by an iterative method, with the goal of finding the distributions that best explain the original data. The assumption about the distributions is the biggest flaw of LDA, which is why NMF tends to give more diverse topics in general.

NMF works by factorizing this term-document matrix into 2 matrices: one that represents the weights of the words for each topic, and one that represents the weight of the topics on each document. The algorithm tries to find which combination of these 2 matrices can capture the original data as closely as possible. The non-negativity comes from the constraint that no word can ever occur a negative amount of times in a document or a topic.

Both of these techniques were applied to the texts, with a varying number of topics and words per topic in order to see which parameters produce the most interpretable results. Both implementations come from sklearn's library. Some topics produced by NMF can be seen in Figure 1, and those of LDA can be seen in Figure 2. Before producing the topics, however, all texts were lemmatized with NLTK wordnet lemmatizer, which was also fed the POS tags from NLTK, for more accurate lemmatization. The lemmatization was performed because, without it, the final topics had words like "says, said, say" (in the same topic), which was of course distracting the users from understanding the true nature of each topic.

```
['  say come know man little make time look think hand friend yes way ask just like door thing quite face good left tell case s
aw old room day said',
 '  bristol halliday carrington man mason know say weston monsieur come daughter jane japp think maid good rupert plymouth trai
n narky yes taunton little make rochefour carriage flossie said return',
 '  giraud jack daubreuil magistrate madame hautet marthe renaulds bella monsieur commissary dagger bex girl villa crime stonor
françoise duveen head merlinville juge beroldy conneau moment dont hastings georges love',
 '  say caroline flora ralph miss blunt parker mr paton raymond know said inspector dont mrs sheppard come im think just minute
ursula ive russell sir fernly ackroyds ferrars dr',
 '  carrington mason halliday monsieur weston sir maid daughter jane rupert japp plymouth good narky taunton rochefour dont cou
nt train mistress carriage mrs jewels poirot return london flossie friend compartment',
```

Figure 1: Some topics from NMF

```
Topic 0:
poirot say know come little man look mr make mrs think friend time yes ask said quite miss tell renauld
Topic 1:
poirot say man come know yes mr hand look head mrs little make thing like room friend door ask quite
Topic 2:
poirot say come know man make little think mr mrs look yes ask time door quite ackroyd just said way
Topic 3:
poirot say man come know make yes time mr mrs little look think said just way ask hand friend miss
Topic 4:
poirot say know come man mr little look make yes time mrs think said hand door ask friend just way
```

Figure 2: Some topics of LDA

It can be seen that the default topics of NMF are indeed better than those of LDA, which is backed up by the theory of the course, which states that NMF produces more diverse topics in general. However, it is not always optimal to look at the first 10 or 20 words per topic in order to understand the topic; sometimes there exists a need to manually pick words from each topic in order to make the topic easier to understand. Therefore, these "manually" picked topics can be seen in Figure 3.

```
['look', 'think', 'time', 'friend', 'door', 'face', 'room']
['halliday', 'mason', 'weston', 'jane', 'japp', 'train', 'plymouth']
['magistrate', 'commissary', 'dagger', 'villa', 'crime', 'judge', 'merlinville']
['caroline', 'flora', 'miss', 'blunt', 'sheppard', 'ursula', 'ferrars']
['maid', 'daughter', 'jane', 'plymouth', 'flossie', 'jewels', 'mistressfriend']
['ursula', 'ganett', 'sister', 'dictaphone', 'secretary', 'blackmail', 'summerhouse']
['strychnine', 'mother', 'poison', 'suddenly', 'hastings', 'wells', 'dorcas']
['moment', 'head', 'truth', 'miss', 'girl', 'mademoiselle', 'question']
['mrs', 'mr', 'cavendish', 'tell', 'little', 'believe', 'tell']
['shook', 'bed', 'remember', 'manner', 'lead', 'star', 'new']
['magistrate', 'comissary', 'francoise', 'villa', 'genevieve', 'buenos', 'ayres']
['hastings', 'dont', 'yardly', 'cried', 'dr', 'wife', 'london']
['lead', 'hercule', 'sir', 'wife', 'insepctor', 'manner', 'understand']
['big', 'chang', 'li', 'savaronoff', 'chinaman', 'countess', 'chinese']
['poirot', 'poirots', 'yardly', 'police', 'minister', 'lord', 'shot']
['prime', 'minister', 'chambermaid', 'lord', 'necklace', 'revolver', 'dinner']
['taken', 'statement', 'facts', 'accepted', 'handle', 'voices', 'nervously']
['examine', 'transparent', 'remark', 'clues', 'admitting', 'erroneous', 'convalescent']
['telephone', 'dinner', 'receive', 'force', 'wall', 'wall', 'visitor']
['measure', 'ground', 'command', 'operations', 'laughing', 'revenge', 'powerful']
```

Figure 3: Manually picked top words per topic for NMF

It can be seen that now the topics make much more sense, and thus they are easier to understand by users.

The next thing presented is the topic river produced for the NMF topics. The visualization is a stackplot from matplotlib visualizing how strong each topic is as the books go from the first to the last (in chronological order of publication), which can be seen in Figure 4

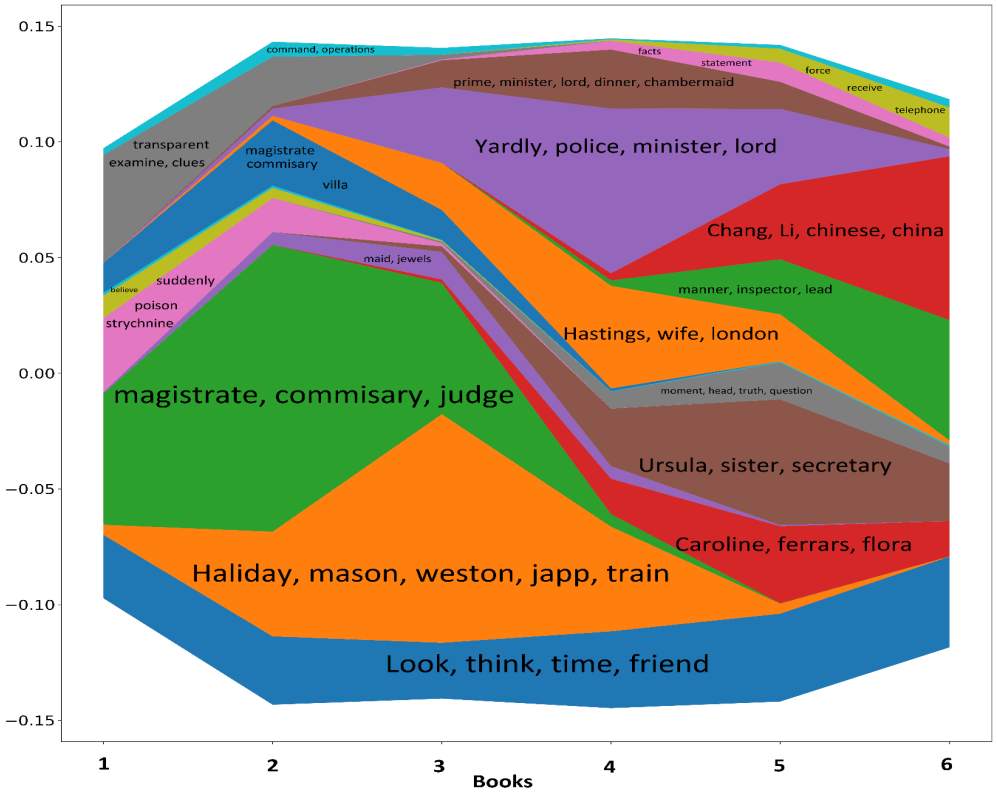

Figure 4: Topic River

It can be seen that there are a lot of topics that are present in all books, while others are present only in earlier books or in later books. For example, topic “think, friend, time” is present in all books, as is topic “measure, ground, operations, revenge”, while topic “ursula, blunt, mrs” is present only on later books, which indicates that maybe the female element of the stories is increased in later books. Topic “magistrate, commissary judge” is present only in earlier books, which indicates that the stories of the first books had a strong element of politics in the background, while later books did not so much. It is now obvious that this analysis can shed light on aspects of the texts that could not be uncovered otherwise.
Table 2 contains the calculated coherence scores for every set of parameters that were tested.

# Topics	# Top words	LDA coherence score	MNF coherence score
20	7	0.2142	0.4148
20	15	0.2884	0.4529
20	25	0.2311	0.4592
10	7	0.4285	0.4144
10	15	0.2840	0.4449
10	25	0.2760	0.4492

Table 2: Coherence score analysis for parameters “number of topics” and “number of top words per topic”

From the table above, it can be seen that NMF outperforms LDA in all but one case. Additionally, 10 topics and 15 words per topic seems to be a good balance between readability of topic (# top words) and explanation of the themes (# topics). The absolute values themselves are promising, but they indicate that the extracted topics are not perfect.

Co-reference handling

In order to track a character’s sentiment or the characters themselves, it is important to first locate the text where they are mentioned. This means that wherever a character is mentioned with a pronoun, the algorithm is going to miss that sentence as a reference to the character, and thus the true sentiment of that character and their path through the story may not be unveiled properly. Therefore, it is deemed important to perform co-reference handling before the sentiment analysis and the character tracking.
The first way to perform co-reference handling was through Spacy and the Neural-coref package that is developed for it. It is based on the neural net scoring model by Clark and Manning [3] (2016), and provides an easy way to both handle coreferences and return the text with all the coreferences handled.

The second way to perform co-reference handling was through FastCoref by Otmazgin et al. [4] (2022). This was also easy to use, but proved to be a little more difficult to return the original text back to the user.
Both methods require chunking the text because of high memory consumption.The two methods were measured only on a qualitative level for their co-reference clusters, since there is not a really good metric to measure performance in this step, and were found to be almost the same in terms of performance. Thus, the resolved texts were used in the sentiment analysis and the character tracking sections.

Character Sentiment Analysis

As mentioned earlier, character sentiment analysis was considered a particularly important and necessary objective to unmask potential suspects and shed light to hidden patterns. This is done by capturing the overall sentiment exhibited by the most important characters throughout the storylines and determining the most important words that influenced specific sentiments. By visualising the behaviour of the top 10 frequently occurring characters per book and considering key words associated with suspicion, the aim is to provide plausible scenarios where, under traditional reading, would not be possible to come up with.

To make the presentation more concise and visually appealing, the focus was chosen to be on one particularly intriguing murder case, namely “The Murder on the Links”. As the title suggests, the story revolves around a murder. By focusing and analysing the negative sentiments associated with each character (with the help of Sentiment Intensity Analyzer of NLTK aslo referring as SIA), potential spotting of the suspect and other deeper insights into the mystery could be uncovered. Below (figure 5) are depicted 8 potential suspect characters with their corresponding sentiment (Negative).

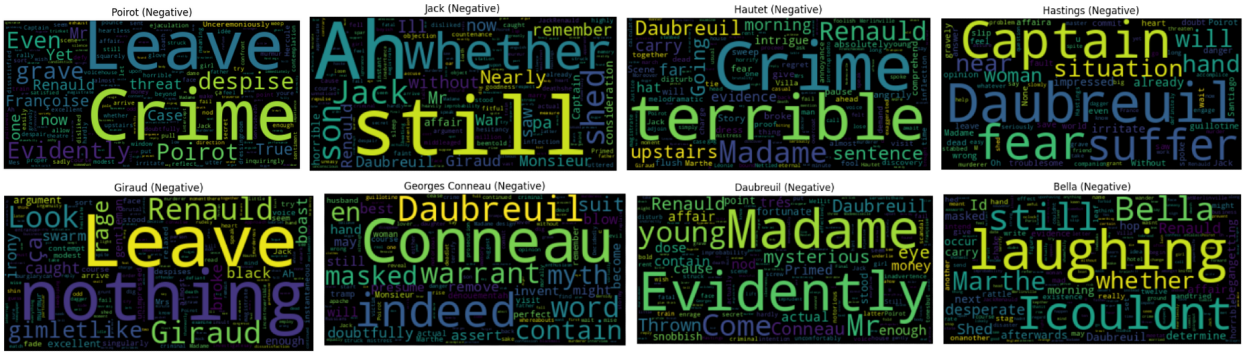

Figure 5: Sentiment analysis for potential suspects

The words represented by each character are exposed words that are contributing to the negative sentiment. The magnitude of words represents the strength or the intensity of the sentiment associated with those words. If a negative sentiment associates with a name it potentially means that the name itself is frequently mentioned in a negative context or

associated with negative sentiment. This could be due to various factors such as negative experiences with characters bearing that name or any other association. Our goal is to expose potential words that could be associated with an underlying murder mystery and to come up with a possible scenario of revealing the murderer.

First of all, it can be easily spotted by some high magnitude words that indeed a terrible crime has been committed(both “Poirot” and “Hautet” referring to crime). In addition to this, words such as “fear”,”suffer”, “terrible” and “rage” enhance the crime. By looking at the most frequent occurring characters within each textural reference, one can easily spot that names such as “Daubreuil” and “Renauld” exposed by most of the characters. Specifically, character “Daubreuil” has been exposed by 5 characters namely “Jack”, ”Hautet”, ”Hastings” ”Georges Conneau” and “Bella”. On the other hand, character “Renauld” has also been targeted by 3 characters namely “Hautet”, “Daubreuil” and “Giraud”. In this case, one perspective says that the more exposed a character is, the more suspect he/she is. Just by considering the above evidences one can express a high susceptibility on “Daubreuil” for the murder of “Renauld”. In addition to this, characters such as “Hastings” and “Hautet” are exposing some suspect words like “terrible”, “fear”, “suffer” ,”crime” and they are also vividly referring to “Daubreuil”. Overall, even from this small lexical representation one can come to extremely important conclusions. There are of course several scenarios that could come into mind and some of them will be discussed as limitations later on.

In the line graph below (figure 6), the overall behaviour of the most frequent character amongst the storyline name “Poirot” can be tracked, according to SIA.

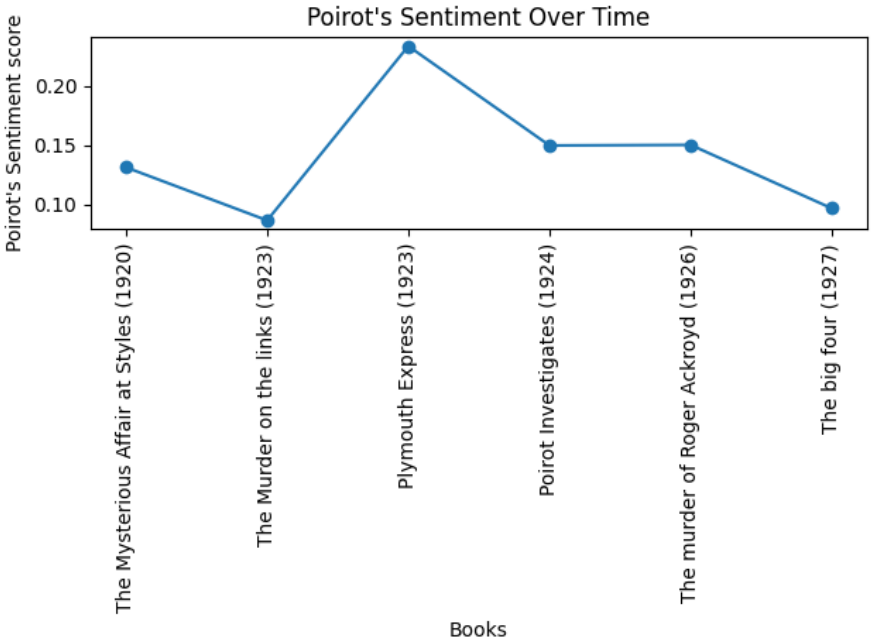

Figure 6: Poirot's sentiment over time according to SIA

The overall sentiment of “Poirot” throughout the novels is positive with some small fluctuations.This could potentially be explained due to some murders. In the book “The Murder on the links” the sentiment behaviour of “Poirot” decreases. After possible

investigations and suspect expositions the sentiment increased while after some other mysterious murders and unrevealed cases fell again. All in all, the main character maintains a positive aspect for the fact that he might hold a deeper ability of what is right and what is wrong and not to easily point into wrong conclusions.

The bar chart (figure 7) depicts the overall sentiment of the 10 most frequently occurring characters per book. The y-axis represents an overall score assignment per each character. Scores below zero represents a negative sentiment, while scores above zero highlights a positive sentiment. Scores close to the dash line correspond to neutral sentiments.

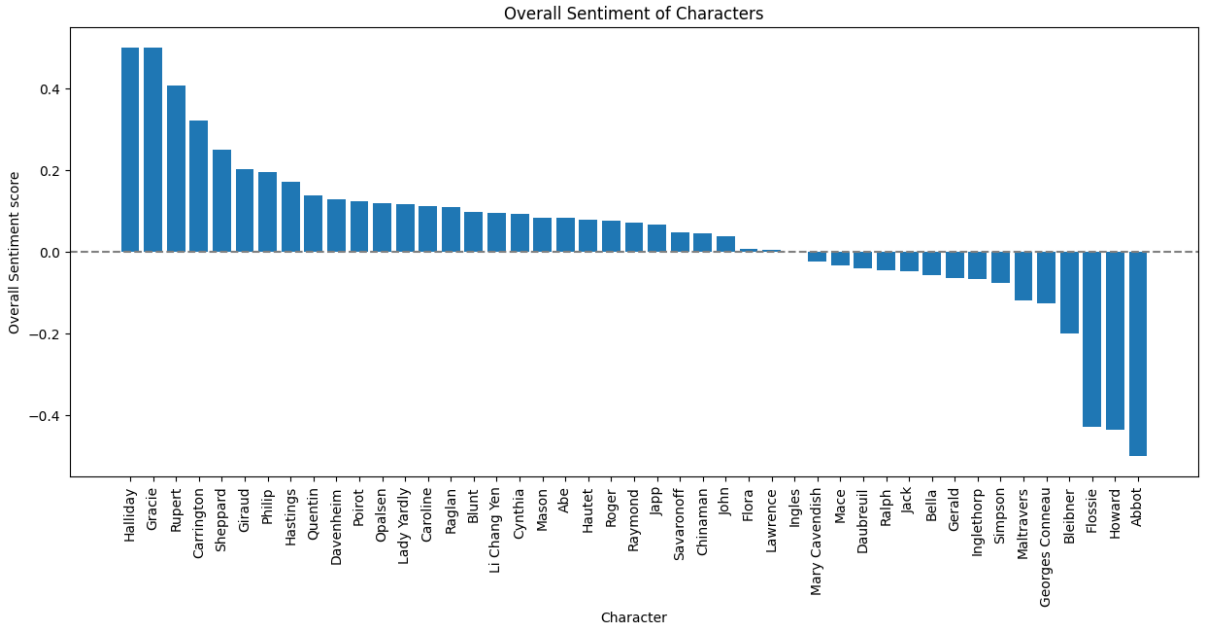

Figure 7: Overall character sentiment according to SIA

Sentiment analysis of main characters through the books

One of the main focus points of this assignment was to analyze the sentiment of the main and recurring characters through the books. This is especially important because it highlights the overall mood and feel of the books, since, if the main character is sorrowful in every scene, then the reader will probably have negative sentiments too. However, if the main character is mostly joyful, then even if the story revolves around the murder, it creates some excitement for the reader to delve deeper into the story. In order to discover the main and recurring characters, the results from the Named Entity Recognition have to be used. The search here is for the tag “PERSON”, its count in every book and if this is higher than 0 (which means that the specific person is present in all books). This happens for Poirot, Hastings and Japp. The next step is to keep each sentence that refers to that specific person in a list, and analyze its sentiment. The sentiment analysis here was done with a pre-trained BERT model, namely crcb/autotrain-isear_bert-786224257 from hugginface, which is trained on 7 sentiments: 'anger', 'disgust', 'fear', 'guilt', 'joy', 'sadness', 'shame'. As the books are in

chronological order, the analysis will reveal how each character evolves through the books with regards to their sentiment and their feelings.
Each visualization reveals the relative power of each sentiment for a specific character up until that point, for every point in the books. This means that if a character is completely joyful in 1 book, but completely guilty in the other 5 books, then in the end, their relative power will be 1/6 joy and 5/6 guilt. Figure 8 contains the visualization for 3 recurring characters, and 3 characters present only in a single book. The top row has 3 recurring characters, and the second row has 3 characters in single books. Figure 9 contains some explainability examples for text sample sentences, with the help of LIME.

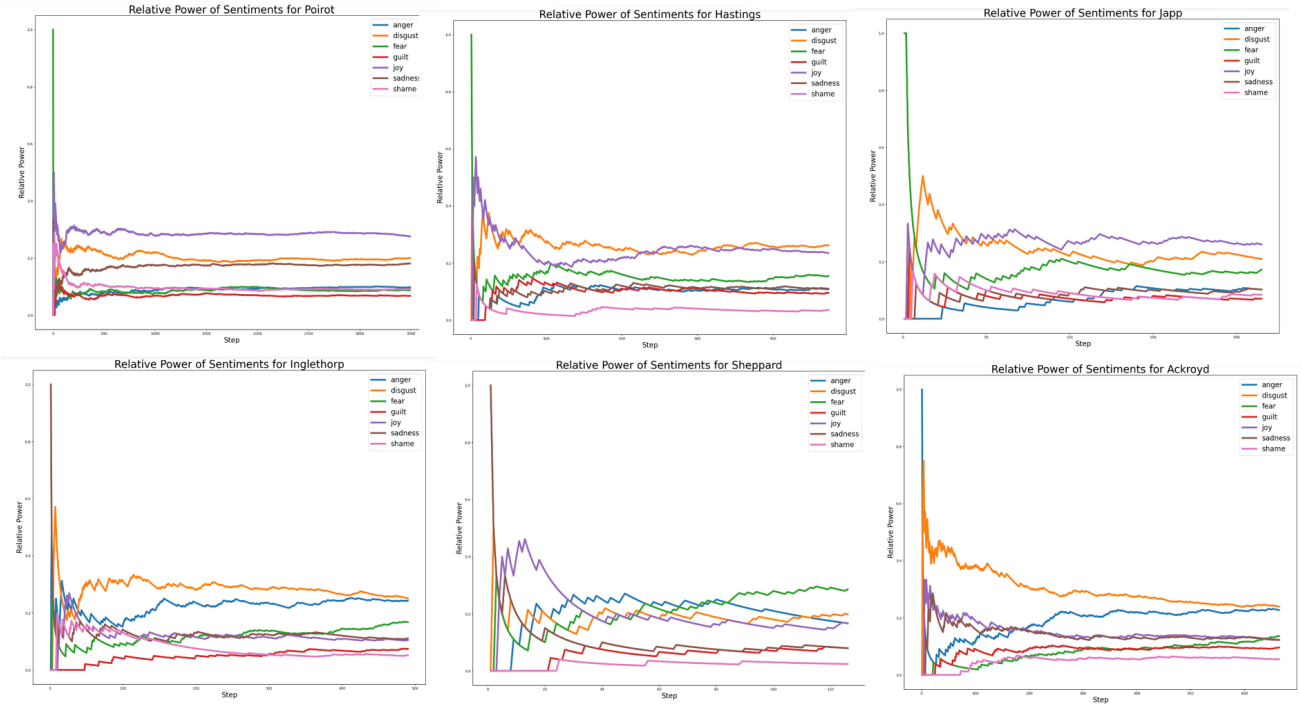

Figure 8: Sentiment analysis on all books according to BERT

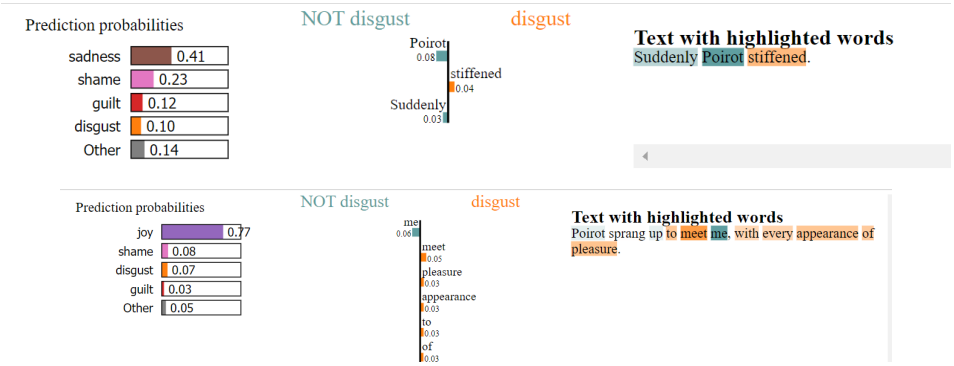

Figure 9: LIME explanations for sample sentences

It can be easily seen that the main character Poirot and his friends Hastings and Japp have overall joyful sentiments, both throughout the books and in the end. Poirot seems to be the happiest in general while Hastings is somewhat disgusted in the middle of his arc. All three

are much more joyful and positive in general than the characters that appear in single books. Inglethorp is a suspect and as such, his sentiment of disgust is rather high (probably towards Poirot), while Ackroyd (the family or Roger) are probably disgusted at the murder of their relative and angered at the suspects and the killer. A lot of family members also tend to feel guilty, probably because they could not help their relatives more.
The fact that the main characters are joyful despite the fact that they are witnessing and solving crimes, is probably the main reason that these books are very well received by the audience. If the main characters are always gloomy and sad, then the reader will probably feel sad as well, and potentially lose interest in the story.
It should be noted, however, that the team felt that there is no point trying to predict who is the murderer in the end, because firstly, these murder mysteries are designed to be especially hard to predict, and secondly, that the classification problem that would arise would be imbalanced: from the 30 or 40 characters in a book, only 1 or 2 are the criminals in the end, so the problem would be highly imbalanced and very difficult to solve. However, from the sentiment analysis, it is easy to see that the suspect pool can be reduced significantly by ruling out happy characters, and focusing more on those that are angry and disgusted (which is probably why they commit a crime).

For the quantitative results part of this section, 40 different sentences were picked and annotated by the team in order to compare to Sentiment Intensity Analyzer and BERT. The results can be seen in Table 3 for BERT, and Table 4 for SIA.

Sentiment BERT	Precision	Recall	F1
anger	1	0.33	0.50
disgust	0.60	1	0.75
fear	0.40	0.50	0.44
guilt	0.33	1	0.50
joy	0.79	0.65	0.71
sadness	0.44	0.80	0.57
shame	0.67	0.29	0.40

Table 3: Sentiment quantitative results for BERT

Sentiment SIA	Precision	Recall	F1 Score
Negative	0.78	0.44	0.56
Neutral	0.44	0.57	0.50
Positive	0.68	0.88	0.77

Table 4: Sentiment quantitative results for SIA

The Cohen kappa score in this section ranges from 30-65% for BERT, because multiple iterations of the annotations were performed since the score was deemed to be low. The main reason for this is probably the wide range of emotions for the particular BERT used here and the fact that most of them are negative (there is only one positive sentiment and 6 negative ones, and no neutral one). Had there been a neutral sentiment for that particular BERT along with the other 7 emotions, the results would have been much better, since the differences between the 2 annotators are in sentences that do not have a particularly strong sentiment, so they should have been labelled neutral.
For SIA, the Cohen kappa score again ranges from 45% to 70%, because of multiple iterations of annotations. The main differences between the 2 annotators are in negative and neutral sentiments. These sometimes are mixed up because annotators disagree if the sentiment of the sentence is negative or the sentiment of the character is negative.

Character tracking

The character that is most important to track in this assignment is of course Hercules Poirot himself. In order to do so, all location tags from the NER section are considered and added to the final location list only if the name Poirot is mentioned in the same sentence as the location, or one sentence before, or one sentence after. This somewhat ensures that the locations in the final list are not just some locations mentioned in the story, but actual locations which Poirot visits. After producing this location list, the location tags are geocoded with the help of Geopy and the final visualization is shown on an HTML file. Some screenshots of this can be seen in figure 10 for Poirot and figure 11 for Hastings. The complete path can be seen by opening the HTML file located in the repository.

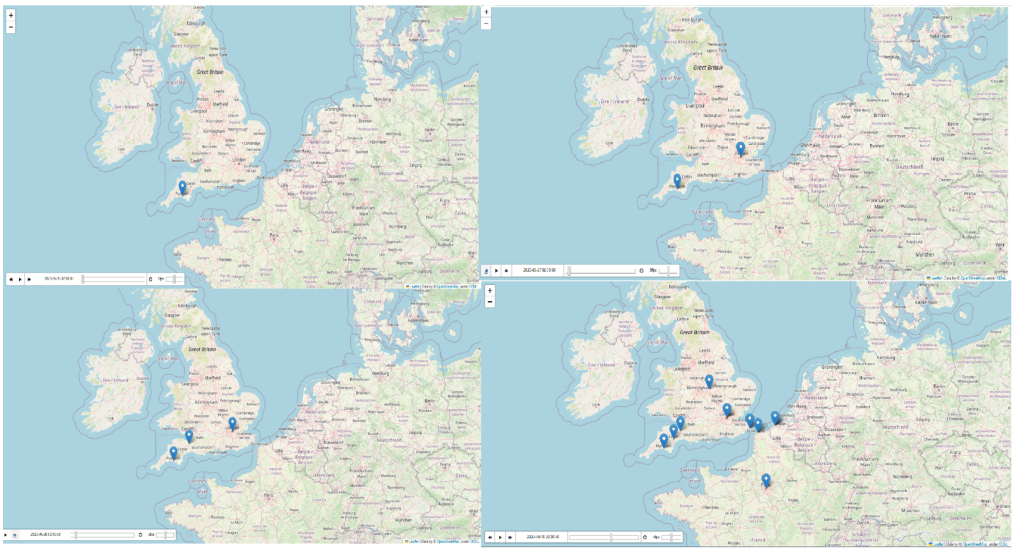

Figure 10: Tracking Poirot through the books

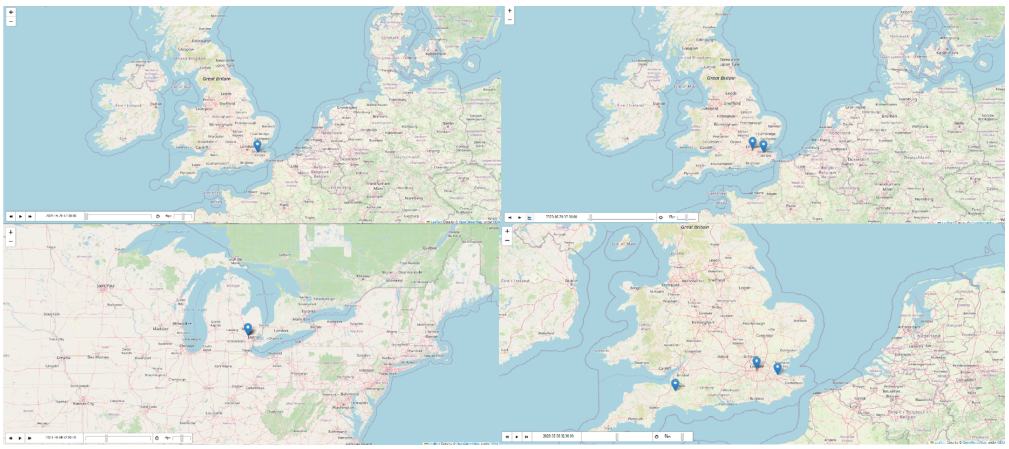

Figure 11: Tracking Hastings through the books

Limitations

This section contains every limitation that was identified by the team for each previous step.

With regards to the text extraction, no limitations were identified, since the text is already extracted.

With regards to the pre-processing:

The team did not read every text from start to finish in order to see if certain special characters still exist. It is possible that some special characters were not properly handled, but these are not showing up in any of the other sections, so they probably do not matter. Moreover, some text is fused together, for example “themotherof” instead of “the mother of”. This is something that cannot be fixed either manually or automatically and decreases the performance of the algorithms used. In another case, there exists a schematic of a map that was shown in the original book, that is added as text in the file. This cannot be deleted without losing some information, so for now it is left as is. However, the team is aware of this issue.

With regards to the Named Entity Recognition:

The most important limitation here is the use of mr and mrs in the text. Of course Mrs Inglethorp is not the same entity as Mr Inglethorp, but when the algorithm identifies Inglethorp as an entity, so in the later stages of the assignment, persons with the same surname may get confused. The way to solve this would be to add a function that checks if Mr or Mrs is the word before each surname, and thus disambiguate what character is being referenced. Again, this has other limitations, such as the fact that Mrs Inglethorp may be referring to the wife or the daughter of a character, so again, this is really hard to deal with effectively.
Other times, Poirot and M. Poirot (not mister Poirot or mr poirot) is recognised as an entity, which again is wrong (the desired entity here is just Poirot).

Locations and GPEs are another thing that gets mixed up in this section, but this is solvable by just replacing both tags with LOC, or searching for both tags when there is a need to find a location.

With regards to the topic modelling and the topic river:

The default top words per topic do not provide a very good picture of the topics to the end users. This means that manual selection of some words was carried out, but this could have led to inaccurate topics. This limitation is a balance between readability of topics and truthfulness to the underlying themes discovered by the algorithm. Another limitation is the fact that “mr”, “mrs”, “say” and some other words are present in almost all topics. This raises the question if these words should be removed from the text beforehand, but maybe that would lead to less accurate results. In any case, this is a limitation that is overcome by just ignoring these words and manually picking the top words per topic.
As for the topic river, the number of topics picked affects the readability of the final plot, along with the fact that the x axis has only 6 values (# of books) which compresses the plot horizontally.

With regards to the co-reference handling:

This step introduces some issues. First of all, the returned text from Spacy sometimes does not make much sense, while the text from FastCoref is especially difficult to return to the user, and also contains some mistakes. This means that some information is lost in this step, however the information gained by replacing (almost) every pronoun that refers to Poirot with the word Poirot is significant compared to the information lost. Besides, the text returned does not really matter for character tracking, since the locations will stay almost the same with the same order, and the only thing that will change will be the pronoun resolution. This, again, is a limitation that is observed and known to the team.

With regards to character sentiment analysis:

The pre-trained BERT model used here should have probably been fine tuned on an annotated novel corpus, but this dataset was hard to find. Another limitation here is the fact that this model does not have a neutral sentiment, while the characters in the text sometimes have a neutral sentiment, creating a loss in performance. This could have been solved in 2 ways: either find another model (which the team could not) or do not consider the sentiments which have a confidence score below a certain threshold. This was decided against by the team because of the potential loss in performance, but it should be added as a future work section.

With regards to the sentiment Analysis for exposing the suspect on the “Murder on the Links” novel:

The sentiment has been analysed over the most 10 frequent characters throughout the presented books. This limitation makes the team question if the existence of a character can

occur if more characters had been considered. Certainly, this possibility is likely, and some important character information may be lost that could lead to wrong discoveries and assumptions or even enhance potential current ones.

It is very important to mention that sentiment visualizations are generated based on the analysis of textural data and mining techniques. While they can provide valuable insights, they might not accurately capture the overall picture and can have some limitations. The negative word correlation that was presented in Figure 5 may not accurately highlight the susceptibility of the murderer. For instance, just by exposing words, it is not certain if the event (textural character occurrence) is taking place as the story unfolds instead of referring to past tense. Furthermore, characters in narratives frequently express irony and sarcasm to convey sentiments. These forms of communication often involve expressing the opposite of what is intended, making it difficult for sentiment modelling to correctly classify and interpret the sentiment. On the other hand, sentiment analysis models require substantial amounts of annotated training data to learn and generalise effectively. However, sentiment-labelled annotations specifically tailored for unfolding murder mysteries in literature might be limited or nonexistent. Last but not least, sentiments can be subjective, and individuals may interpret sentiments differently based on their own book experiences, biases and more, thereby, they lack personal context and subjectivity understanding.

With regards to the character tracking:

The most important limitation here is of course the use of past tenses and the introduction of new characters from locations. For example, there is a certain part in one book that refers to a person from China. Now, of course, when Poirot meets this person, the name Poirot is very close as a token to the token China, which leads the algorithm to the conclusion that Poirot travels to China. Of course, this is wrong and leads to a loss in performance that depends on how many characters are introduced in the novel, and how they are introduced. This could be solved by making some extra steps in the algorithm, such as excluding locations when they are preceded by “born in”, “from” etc. However, this could lead to other inaccuracies as well, so the team decided against it.
The order of the locations could be wrong as well, since if Poirot himself is talking about a past experience in Greece, again the algorithm will deduce that Poirot goes to that particular location in the present, which mixes up the order of locations in the story. This is especially hard to solve, and requires a lot of processing by the team, and is thus left as a future work section.

References

1) Contextual String Embeddings for Sequence Labeling. Akbik, A., Blythe, D. & Vollgraf, R.. (2018). *Proceedings of the 27th International Conference on Computational Linguistics*. https://aclanthology.org/C18-1139

2) Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International Conference on Weblogs and Social Media* (ICWSM-14). Ann Arbor, MI, June 2014.

3) Clark, K. & Manning, C.. (2016). Deep Reinforcement Learning for Mention-Ranking Coreference Models. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. DOI: 10.18653/v1/D16-1245.

4) Otmazgin, S., Cattan, A & Goldberg, Y.. (2022). F-Coref: Fast, Accurate and Easy to Use Coreference Resolution, arXiv:2209.04280.

5) Manning, Christopher D., Prabhakar Raghavan and Hinrich Schütze . “Introduction to information retrieval.” (2005).

6) https://predictivehacks.com/topic-modelling-with-nmf-in-python/

7) Attention is not Explanation, Sarthak Jain, Byron C. Wallace. https://arxiv.org/abs/1902.10186 .