Zhou Lu, Ningjun Li, Huizhirong Guo
CSCI 185
Final Project Report
Dr. Ghosh

Exploring "The Pillars of the Earth" Through Text Mining: A Study of Word Associations and Importance

Background:

The novel's immersive universe is full of patterns, associations, and themes that play a key role in plot progression, character development, and thematic exposition. However, unearthing these patterns and associations often requires meticulous and time-consuming analysis, which can be a daunting task for even the most astute reader. Our research proposes an innovative approach to address this challenge by applying association rule mining (ARM) and word frequency-Inverse Document frequency (TF-IDF) analysis to the text of Ken Follett's historical masterpiece, The Pillars of the Earth.

We focus on the "pillars of the Earth" because of its rich narrative and diverse cast of characters, making it an ideal test bed for our analytical approach. By harnessing the power of ARM, we aim to discover implicit, previously unknown and potentially useful relationships between words, characters and themes in fiction. Essentially, we are looking for patterns such as "if a paragraph contains the word 'cathedral,' it may also contain the word 'builder'" - an association that seems intuitive but quantifies this relationship can provide insight into the author's narrative style.

As a complement to ARM, we incorporated TF-IDF analysis into our study. The ARM reveals interesting relationships, while the TF-IDF helps determine the relative importance of words in the context of the entire novel. By combining these two powerful analytical tools, we can not only discover relational patterns, but also weigh them based on their importance to the overall narrative.

Integrating ARM and TF-IDF into a single novel for text analysis is a relatively unexplored area of research, which makes this research project innovative and potentially influential in the field of computational literary analysis. By delving into this novel approach, we hope to provide deeper and more specific insights into Follett's text that will contribute to a broader understanding of narrative structure, thematic patterns, and character interrelationships in literary texts. We anticipate that our findings will stimulate further discussion at the intersection of literature and data science, promoting a more nuanced appreciation of literary works."

The aim of this project is to gain a deeper understanding of Ken Follett's novel The Pillars of the Earth by applying text mining techniques, specifically TF-IDF and association rule mining. This will involve identifying the most important words in different parts of the text and discovering associations between words that reveal underlying themes, character relationships, or plot structure. As the novel was created under the background of real historical events and the chapters progress chronologically, we expect to extract interesting patterns and connections that reflect not only the narrative structure of the book, but also its historical nature.

Dataset:

We are using the novel "The Pillars of the Earth" by Ken Follett as the primary dataset to perform association rule mining and TF-IDF analysis. The novel is an historical epic, encompassing a vast array of characters, locations, themes, and events that provide rich raw material for our data mining procedures. There is an enormous relationship between the characters and the plot in the novel. By analyzing the dataset by dividing chapters into paragraphs and dividing paragraphs into words, it is efficient for readers to summarize relationships between characters and the stories within each chapter. By viewing the analysis result of the whole book, the readers can easily find out who is the main character of a specific chapter. As a novel with a rich plot, it is suitable for a project like this to help find the meaningful connection between words using data mining.

Tools:

We mainly used different libraries to help us generate the rules and relationships between different characters and plots. We use association rule mining and TF-IDF analysis to analyze each character and each chapter, and generate graphs to help identify the relationships.

Data Preprocessing:

We first reads the PDF book using the PyPDF2 library's PdfReader class. By building an object using the function ‘PdfReader,’ we can extract text from a specified range of pages from the PDF. Then we defined a function ‘extract_text_from_pdf,’ to divide the book into chapters based on predetermined page ranges. By doing this, we can analyze the appearance of characters and plots within each paragraph specifically. After extracting the chapters, we saved the entire book into a single text file for further implementation.

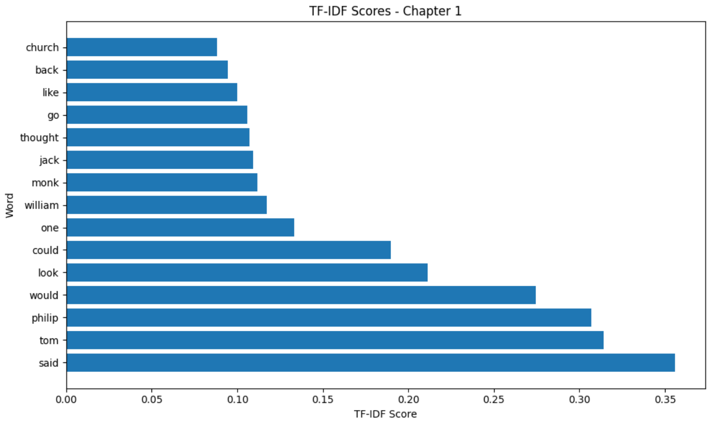
Then we convert all characters in the text file to lowercase to avoid duplicates caused by case differences and calculator errors to provide us with an accurate result. After the lowercase processing, we can use the nltk library to tokenize the words with the word_tokenize() function. As the tokenization is done, we remove stop words, which are common words like 'the' and 'and' that are often filtered out in natural language processing because they carry less meaningful information to ensure the accuracy of analysis.

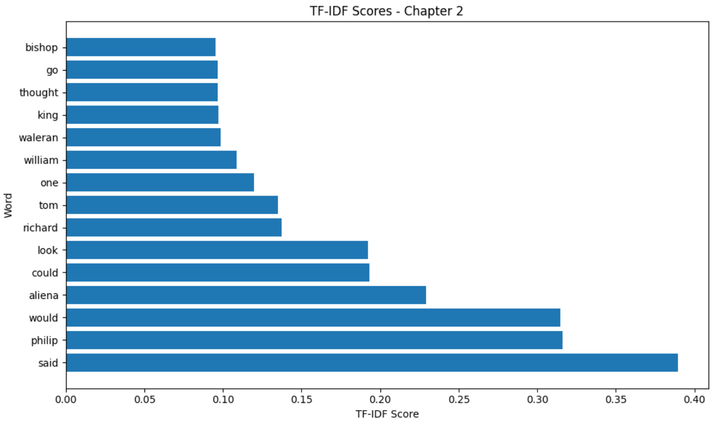
Implementation:

TF-IDF Analysis:

We will outline how we implemented our text mining approach and the methods we applied to ensure we gathered meaningful insights from our text data. To carry out the data manipulation and computation for our text analysis, we used the nltk, pandas, scikit-learn, and matplotlib libraries. The nltk features we leveraged were word_tokenize and PorterStemmer. These helped us to break the text data into individual words and reduce them to their base forms, eliminating variations caused by prefixes and suffixes.

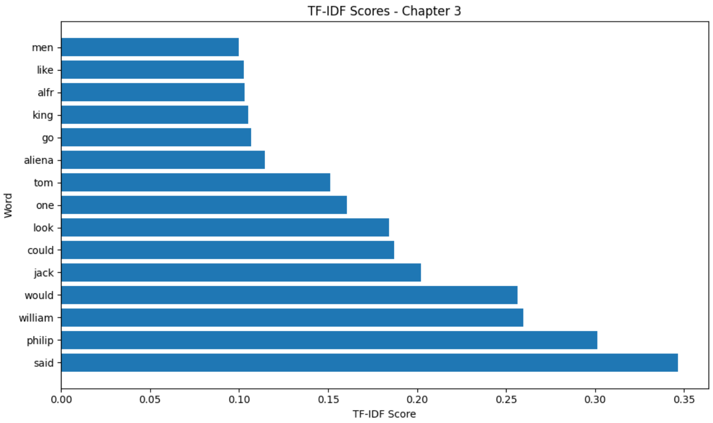
To build the TF-IDF statistic, we used the TfidfVectorizer from scikit-learn to compute the TF-IDF scores for each word in each chapter. To gain a clearer understanding of the distribution and significance of words, we created bar plots for each chapter, showing the top 15 words based on their TF-IDF scores. We were able to observe the most prominent words, giving us an understanding of the story and focal points within each chapter.





As the graph shows, in chapter 1, the graph clearly shows the frequency of different words in the chapter. Tom and Philip has a relatively higher score which clearly represents they are the two most important characters in the chapter, and the most important theme church comes out.

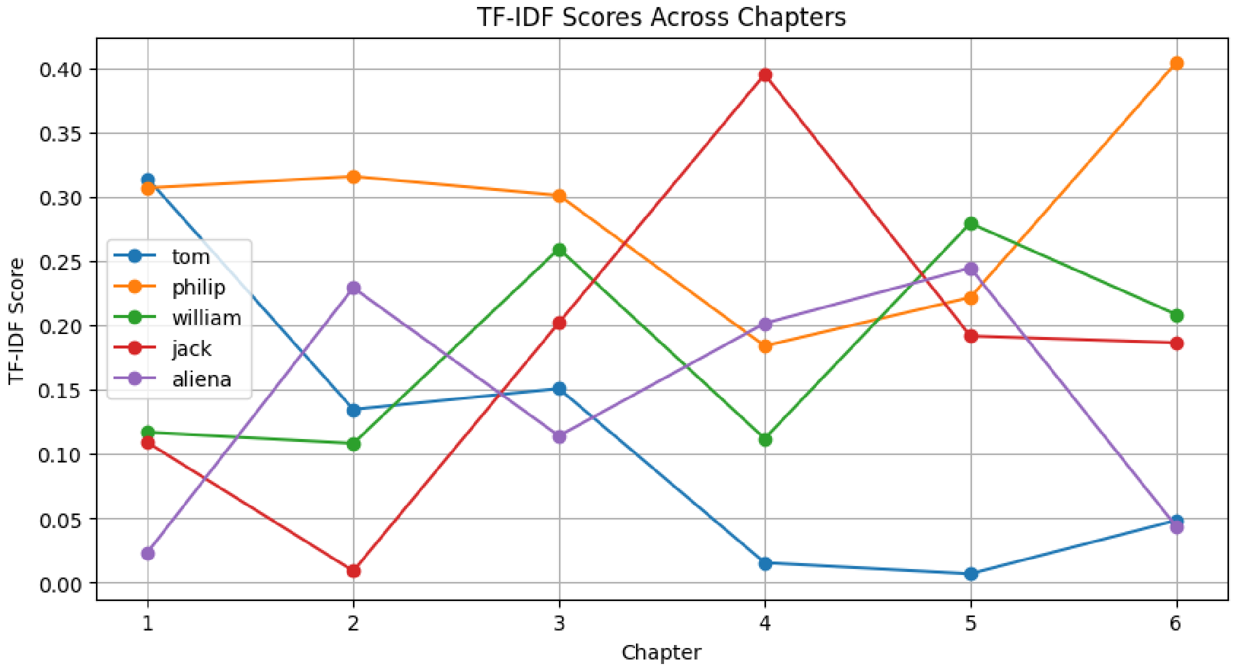
In chapter 2, there shows Philip still got a high TF-IDF score, and there comes out more frequent characters. We can conclude that the story is bringing out more characters and making the story into a broader view, and the relationships between different characters are becoming complex.



The chapter 3 result still concludes that Philip is the character who appeared most frequently, so as the following chapters. Based on the data, we can conclude that Philip's name occurs with high frequency throughout the chapters, implying he plays a consistent and significant role in the unfolding story. His consistent appearance could indicate that he's involved in many events or that the narrative often revolves around his perspectives or actions. It's also plausible that Philip's interactions and relationships with other characters are key to the novel's storyline.

However, we found that it is not clear to see how the plot goes on with separate graphs, so additionally, we tracked specific words across the chapters using line graphs to understand their progression and significance throughout the narrative. The line graphs showed us the TF-IDF scores for each selected word across all chapters, giving us a visual representation of how these words and the characters they stand for were distributed throughout the novel, and get a general idea of how the relationships between characters goes on.

We employ the matplotlib library to generate graphical representations of these scores. The script starts by defining a list of words to analyze. For this instance, those words are the character names, 'tom', 'philip', 'william', 'jack', and 'aliena'.This rigorous approach allowed us to successfully transform the unstructured data of the novel into structured data that could be easily analyzed and interpreted.



Upon visual inspection, the bar plots and line graphs gave us clear insights about the narrative's progression and the characters' development throughout the chapters. In contrast to the TF-IDF bar chart in separate chapters, this graph shows the importance of each character in different chapters more clearly.

In our list, we have five characters: 'tom', 'philip', 'william', 'jack', and 'aliena'. The plot has each of these characters represented by a different line. The x-axis represents the chapters of the book, while the y-axis shows the TF-IDF score, which is a measure of how often each character's name appears in each chapter relative to its appearance across the whole book.

If we notice a spike in a character's lines in a chapter, it's an indication that that character is particularly prominent in that part of the story. For example, "Philip" has a relatively high TF-IDF score in Chapter 2, it means that Philip is a key character in that chapter, possibly due to an important event or part of the narrative focusing on him. Conversely, if a chapter has a low TF-IDF score, it indicates that the character is not an important focus of the story in that chapter.

By observing the trajectory of each line, we can observe the development of each character throughout the book. A steadily increasing line could mean that the character becomes more and more important as the story unfolds. At the same time, decreasing lines may indicate decreasing character importance.

This plot can be used to analyze character trajectories and characters in the story. However, it's important to remember that while TF-IDF scores can indicate a character's prominence, they don't provide information about the context or nature of a character's involvement in the plot. Overall, it is an efficient way to conclude the storyline.

Association Rule Mining

We first use a simple method to explore the context of specific words in a text. We identify words that frequently appear around our target words within a set window of surrounding text. By tokenizing the text into individual words and counting the frequency of words near each occurrence of the target word, we can gain insights into the typical context these target words are used in. Through repeated experiments, I finally set the window size to 20, which means recording the number of occurrences of all 20 words around every time the target word appears. We also remove the stop words in this method to make the result more straightforward and clean.

This method helps us understand the narrative or discussion associated with the target words. One advantage of this method is that it is highly efficient and quick to execute. From the top 20 words associated with each target, we can extract substantial insights into the character identities and their relationships with others. For instance, in Philip's list, 'prior' frequently appears, signifying his title. Following 'prior' is 'waleran', which hints at a rivalry or significant interaction within the church. The presence of words like 'bishop', 'monks', and 'church' further suggest Philip's affiliation to the clergy and his primary sphere of activity being the church. Thus, even from a cursory analysis, we can deduce Philip's religious vocation and his key associations within the novel.

However, the problem with this method is that it does not directly provide information about relationships or associations between words. So, in the next stage of our analysis, we plan to use the Association Rule method. In our Association Rule mining method, we also begin by tokenizing the input text into sentences or paragraphs using NLTK's sentence tokenizer. We then further break down these chunks into individual words while simultaneously eliminating stop words, such as "the", "is", "at", etc., which bear limited semantic value.

In our advanced textual analysis research, we've employed the Association Rule mining method to uncover intriguing connections between specific target words and the rest of the text. This method begins by tokenizing the input text into sentences or paragraphs using NLTK's sentence tokenizer. We then further break down these chunks into individual words while simultaneously eliminating stop words, such as "the", "is", "at", etc., which bear limited semantic value. Once we have the list of filtered, tokenized sentences, we apply this approach to each of our target words. Importantly, we now consider all sentences, not just those in the vicinity of the target word. This is a critical aspect of Association Rule mining, as it provides a more comprehensive overview of potential associations in the entire text.

We then transform these transactions into a binary encoded matrix using the TransactionEncoder from the mlxtend library, enabling our dataset's compatibility with the Apriori algorithm. The matrix is converted into a pandas DataFrame, an efficient and flexible data structure ideal for the subsequent computations. Applying the Apriori algorithm on this DataFrame, we carry out frequent itemset mining to discover sets of words that frequently occur together. After determining the frequent itemsets, we proceed to generate the association rules, which identify pairs of words that consistently appear in the same context, and thus may share some semantic relationship.

The critical part of our approach is filtering these rules to include only those associated with our target words. For each rule where the antecedent is a target word, we store the confidence and the lift. The confidence measures the rule's reliability, while the lift indicates how much more often the antecedent and consequent occur together than we would expect if they were statistically independent.

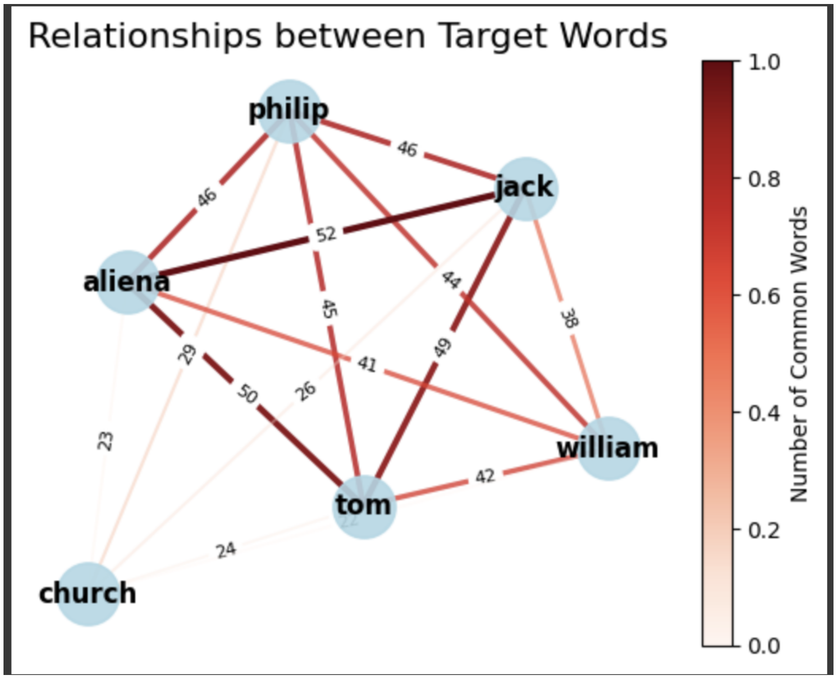
The result is a detailed list of words most strongly associated with each target word, based on their confidence and lift. We sort this list by confidence and take the top 20 associated words for each target word. Finally, this data is compiled into a pandas DataFrame, offering a clear, tabulated view of the significant associations for each target word.

However, when applied to large texts, like a book, the Association Rule mining method does have its challenges. Given the vast number of words, the support between two words can be very small, which can result in an overwhelming number of rules that may not all provide meaningful insights. Moreover, the computational cost for generating these rules can be substantial, making the method time-consuming. Within the threshold I set, the program ran for about 15 minutes, and if I continued to lower the threshold, the time would increase exponentially.

An example of this limitation can be seen in the analysis of the word "church". Despite being one of our target words, no association rules involving "church" met the thresholds set for confidence and support in our Apriori algorithm. This illustrates how the method can sometimes fail to produce useful rules for certain words, particularly when dealing with large datasets. The only target word that returns all of 20 words is "Philip". This represents Philip's status as the absolute protagonist of the book and in connection with other characters and events.

Combining results from two methods, we get some really interesting findings. Association Rule Mining confirms the relationships between words obtained by the previous method and also gives us additional information about these relationships, such as the strength, direction, and significance of the associations. By evaluating the confidence and lift of each association rule, we can identify not only which words (or in this case, characters) frequently appear together, but also gauge the likelihood and intensity of their co-occurrence.

One example is that in aliena's windows, “richard” and “jack” appear with similar frequency, but Jack's lift is substantially lower than Richard's. This demonstrates that while Jack and Alien often appear together, Jack's significance extends beyond his connection with Alien. As a central character, Jack has profound ties with numerous other figures. In contrast, Richard, as a supporting character, primarily interacts with Alien, which results in his comparatively higher lift. Therefore, despite similar occurrences, the interaction dynamics differ vastly for Jack and Richard in the context of Alien's narrative.



The graph above performs text analysis to identify common words surrounding target words in a given text. It demonstrate a relationship graph among characters based on these common words. The main steps include extracting nearby words around each target word, finding the common words shared between different target words, and visualizing the relationships using a graph. The code that used to implement this graph starts by defining a function that tokenizes the text and retrieves nearby words within a specified window size around the target word. It then counts the occurrence of each word and returns the most common ones. Next, the code iterates over a list of target words, calling the function for each word to obtain the common words associated with it. It stores the results in a dictionary. The code then finds the intersection of common words among all target words and removes these intersecting words from the results. This ensures that the relationships captured in the graph are unique to each character. Using the network library, the code creates a graph and adds nodes corresponding to the target words. Edges are added between target words based on the shared common words, with the edge weight representing the number of shared words. The graph is then visualized by drawing nodes as light blue circles and edges with varying widths and colors based on their weights. Edge labels indicating the number of common words are added. Finally, the code displays the graph plot with a colorbar representing the edge weights. In summary, this graph provides a way to analyze and visualize the relationships between characters based on the common words they share in a given text.

Conclusion:

In conclusion, our exploration of text mining techniques, particularly Association Rule Mining (ARM) and Term Frequency Inverse Document Frequency (TF-IDF), applied to Ken Follett's novel "Pillars of the Earth" generate a profound result.

While the TF-IDF analysis revealed the frequency and importance of characters in each chapter, the ARM brought to light potential relationships between characters and themes in the narrative. The association rules shed light on significant associations between the target words, revealing a broader perspective of the text beyond the immediate vicinity of these words.

Our findings suggest that the two techniques can complement each other well. ARM reveals hidden relationships in the narrative, while TF-IDF provides a numerical measure of the importance of these relationships. By combining these two approaches, we were able to gain a deeper and more comprehensive understanding of story structure, character development, and thematic content.

This innovative methodology can be applied to other literary works, providing a more nuanced approach to literary analysis. We hope this kind of analysis method can be an innovation to the literature field. By combining other quantitative and qualitative methods, we can deepen our understanding of complex narratives such as "Pillars of the Earth" and make a significant contribution to other computational literature analysis.