Social Data Analysis and Visualization of *A Song of Ice and Fire*

An analysis and visualization assignment using Python programming language.

Thea Rasmussen (s113611)
Jacob Lambert (s136215)
Pernille Bernth (RUC stud)

ABSTRACT

In this study, we analyze the characters from George R.R. Martin's *A Song of Ice and Fire* book universe. Specifically, we want to use various methods of text and data analysis, as well as visualizations, to detect patterns in relationships between characters. Our main methods consist of word frequency visualization using word clouds, character frequency visualization by using a linear timeline graph, static and dynamic social network construction to visualize relationships, and a Naïve Bayes machine learning classifier to predict what these patterns and relationships may implicate about the future of the series. Although our study focuses on a fictional universe, the methods used can easily be generalized to a real-world data set.

1. Motivation

Our study attempts to apply methods commonly used in social network analysis to a fictional universe. We have chosen George R.R. Martin's *A Song of Ice and Fire* for several reasons. First, the books' complex and often intertwined relationships between the large number of primary characters leads to a very interesting case study for social data analysis. Also, there exists a wealth of external fan-submitted data about the series on the internet wiki, ww.asoiaf.westeros.org. We made several decisions about character choice and relevancy based on data scraped from this source. Additionally, the considerable length of the five released books in the series provided a large data set for analysis. Finally, because the series has not yet concluded, we aimed to make interesting and relevant predictions about the future of the characters, and due to the popularity of the television adaptation, *A Game of Thrones*, our findings will be relevant to a general audience.

2. Theory

The theory behind our methods included simple yet meaningful visualizations, static and dynamic network analysis and visualization, and a machine learning component using a Naïve Bayes classifier.

2.1 Visualization

Our simple visualizations included word clouds and a character timeline for each book. We chose to create word clouds to show major themes throughout the series and how they change over time. The character timelines we created show the presence or absence of the most important characters through the books, which shed an interesting angle on how the author chose to develop the characters. Also, we can see how the primary protagonists and antagonists change over time.

2.2 Network dynamics

Our social network analysis consisted of both creating static networks for each book and a dynamic network for the series. The static networks effectively summarize the relationships throughout each book. We can see which characters were most strongly related and how these relationships may have affected the well-being of the character. The goal of our dynamic networks was to visualize these relationships on a local scale, and observe local patterns that could be overlooked in the larger scale of the static networks.

2.3 Naïve Bayesian Theory

Finally, we chose to use a Naïve Bayesian classifier to attempt to learn about the word patterns surrounding certain characters. We first train the classifier using sets of words around know negative characters, or characters who experience misfortune or death , as well as sets of words around known positive characters, or characters who are successful and alive. We then apply these learned patterns in order to make predictions about the future of other more neutral characters.

3. Implementation

Before conducting and analysis or visualization, we first needed to obtain screen-readings of the books. Once we had converted the .pdf screen readings to text files, we used the command-line regular-expression utilities grep and AWK to filter out punctuation, stop-words, as well as standardizing capitalization. Finally, we used the community wiki to determine relevant aliases and nicknames for the characters, and replaced them with the characters' names using the sed stream editor.

3.1 For the implementation of our word clouds, we heavily relied on the freely available online application, at www.tagxedo.com. After creating the text files for each book, we were able to directly upload documents to the application and display the results. In our character appearance timelines, we exported data from python and used Javascript for visualization. The x-axis is an indexing of the words in each book. For each character, we denote their appearance at an index in the book by a colored bar.

3.2 The construction of the social networks consisted of several steps. We created a static network for each book using the python module NetworkX. The nodes in our networks represent the most important characters, which we determined using the assistance of the online source and personal authority. The edges in the network represent a connection between characters, which we quantified by the appearance of the names of both characters within 50 words in the formatted text. We chose this quantification because we felt it most often represented situations where two characters simultaneously appeared. The weight of our edges represents the number of associations.

We used the open source application Gephi to import networks from python and Networkx. In the static visualizations, the node size is proportional to the degree of the node, and the color represents the demographic of the character. For characters in Westeros, this most often represents their surname or house. Otherwise, it could be their clan or associated city. Finally, the edge thickness is proportional to edge weight.

We also used Gephi to import our dynamic social networks from Networkx. The attributes of the nodes and edges represent the same information as the static graphs. However, due to import complications with Gephi, our networks do not accurately display the dynamics of the network. To correctly import these networks requires an intermediate text-processing step which we have yet to implement at the time of this writing.

3.3 Our Naïve Bayes classifier used a class construction similar to that presented in T. Segaran's *Programming Collective Intelligence.* We denoted documents by each set of 20 words surrounding a character, and the document features as the individual words. After training the classifier using the set of all documents surrounding 4 positive and 4 negative characters, we began to classify other characters. For each character, we created a list of documents based on the 20 surrounding words each time they were mentioned. Then we classified these documents, and created a "positive / positive + negative" probability for the character. The results of this analysis can be seen in the website's scatterplot.

4. Discussion

Our data analysis and visualization has uncovered many interesting results, as well as additional avenues for exploration. However, there are several considerations for method choice and implementation that require further research and development.

4.1 The word clouds and timelines give a general overview of the books from interesting angles. While most often the dominant words in the clouds are names and titles, often some unexpected words garner attention, such as "hand" in *A Dance with Dragons*. This sheds light both on the importance of the position of "Hand of the King", as well as the amputation of the hand of a point-of-view character, which are likely related metaphorically. With our timeline plots, we can see interesting development with the characters. The presence and absence of the characters is heavily influenced by the point-of-view writing style of the books, where one character is the focus of each chapter. However, even with this style the influence of certain characters, even after their death, in other characters' chapters is evident, for example Eddard Stark in *A Feast for Crows.*

4.2 Our static networks show many interesting elements of the series. We can see which characters and houses are closely connected, as well as subplots and character presence and absence. However, alterations in method design could lead to different results in these networks. For example, we could have chosen different characters to analyze, different metrics for associations, or different demographic classifications. These alterations are interesting for future research.

4.3 Finally, the results from our Naïve Bayes classifier are largely inconclusive at this point. Although we had access to a large data set, we discovered that our choice of training data was susceptible to bias and created significant differences in the classifications of the other characters. Also, the formulas used for weighting the probabilities of the features and documents caused the classifications to be heavily influenced, not by the words surrounding a character, but the presence of characters used in the training data. However, we feel that additional development and testing of this method could lead to decisive and conclusive results and would be an interesting direction for future research.