Malazan Network Analysis

Christopher Peralta

Friday, Apr 26, 2019

Abstract

Note: All bolded words are hyperlinks.

In this project I attempt to find the climaxes of the series **Malazan Book of the Fallen** by using network data and sentiment analysis. This series is notable in that it is one of the most complex and long fantasy series with a continuous single plotline. There are 3,252,031 words in the series and I estimated that there are 1,479 characters in the series with approximately 457 unique points of view. Additionally, many of the characters have multiple aliases and nicknames adding another layer of complexity. A character might go by completely different names in different novels.

I began by mining the co-occurrence data from the ngrams; I used a combination of regular expressions, parallelization, and more. I'll go more into that process later on. From there, I had to get the co-occurrence data into a reasonable format and clean the name data. I then used the AFFINN Sentiment Lexicon to get the sentiment data. Finally, I compared the sentiment data with the network data to try to see which works better to determine the climax of each of the 10 books.

My general conclusion was that edge betweenness centrality aggregated by chapter using the mean does the best job of finding the actual climaxes of each book.

Introduction

The goal of this study was to use network and sentiment analysis to find the climax of each book in the series. I used numerous datasets from several sources in this project. The co-occurrence data was mined from the books by me. The books were converted from `.epub` format to `.txt` format. I made most of the `alias` data manually and crowdsourced some of the aliases on **Reddit**. The `name` data was manually extracted from the *Dramatis Personae* sections at the start of each book and manually extracted from the **Malazan Wiki**.

To mine the character co-occurrence data, I started by converting `epub` versions of the novels to `.txt` files and reading them into R. Then I added book numbers, chapter numbers, and stripped the front and back matter from each book. From there, I turned the text into a series of ngrams of length 20 split by book and chapter. At this point, I wrote a function that extracts the characters' names from the ngrams by row and then puts the co-occurrence data into a workable format. Cleaning and formatting the co-occurrence data was the next step. Finally, I used a variety of methods to find the climaxes of each book.

Method and results

Method

I'll begin this section with a detailed description of the methods and assumptions I used in mining the character names from the novels as that was the most difficult part of the project.

I began by joining the character name data with the alias data into a single dataset. I then split all of this data by spaces in order to get variations of the names and rejoined the partial name data back to the full name data to get a comprehensive dataset of full and partial names. I then filtered out stop words, formal titles, military ranks, and commonly capitalized words that aren't names from this list. Then, I arranged the list by character length.

At this point, I went back to the book data and turned the text into ngrams of length 20 by book and chapter. I chose to use ngrams so that I would get the full co-occurrence relations within the 20 word groups. For example, "Ron Jon" then "Ron Jon Bob" and finally "Jon Bob". Then I wrote a function that tries to extract every name in the name list from the ngram. If there is a match, then it also removes the match from the string for subsequent iterations. Otherwise, "Brys Beddict" would be extracted 3 times.

This method of extraction was incredibly computationally intensive as there were 3,080 names in my name list and 3,250,530 ngrams. My code went through multiple iterations, and I eventually added parallelization and broke my data into 263 chunks. All of this managed to get my code to run in around 40 hours on my laptop. The original speed was about 310 seconds per 1600 ngrams, and I got that down to about 70 seconds per 1600. These speeds are what I recall, initially I did not think to record them. Most of the improvement was due to the parallelization and an `if` statement before the `str_extract_all` call. I attached the code for this part in the appendix.

Once I got the co-occurrence data into a usable format, I had a lot of data cleaning to do. A lot of the data cleaning was due to the fact that there were partial name matches due to the use of ngrams "John Smith" matched as "John", and due to the fact that some characters have up to 9 different names that they might go by. My job was made slightly easier by the fact that less important characters tend to have fewer names in the series, such as Mallet, Picker, or Antsy whose surnames are never revealed. I formatted and removed variations of all names with over 100 appearances in the network data. I used over 130 regular expressions to achieve this. I made three important assumptions at this stage:

- The names with over 100 occurrences in the co-occurrence data in addition to the uncleaned names of the less featured characters are sufficient to fully represent the true co-occurrence network.
- The most common variations of names accurately represent the co-occurrence relationships of their specific character.
- Any extremely uncommon name variations will be filtered out as isolated nodes or by a small filter removing the most uncommon nodes.

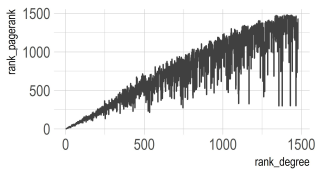
After cleaning the data, I began to analyze the co-occurrence data.

Results

I'll jump right into it by giving the top 10 most important characters by PageRank compared with their degree centrality importance. I only used undirected network graphs for everything that follows.

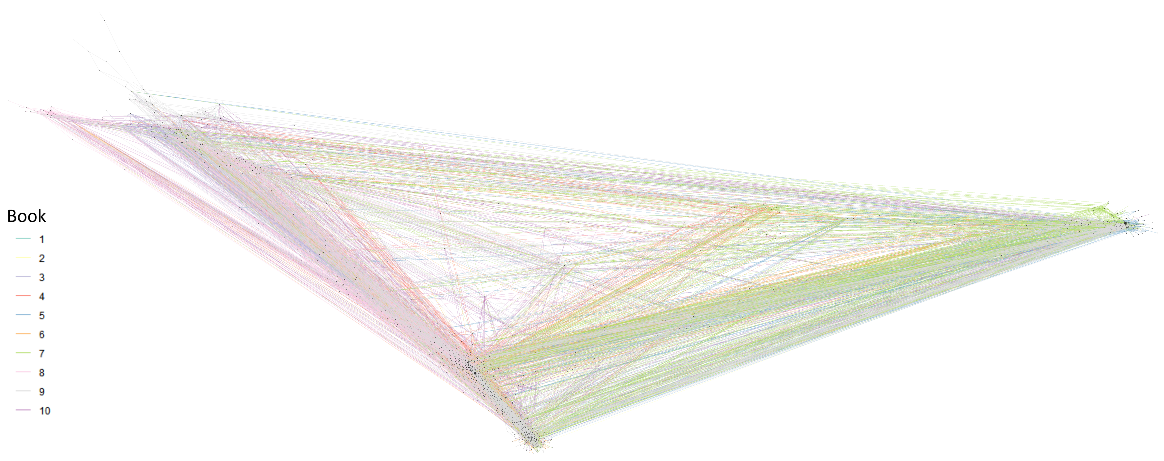
Character	Ranking - Degree Centrality	Ranking - PageRank
Tavore Paran	1	1
Ben Adaephon Delat	2	2
Fiddler	3	3
Hood	5	4
Ganoes Stabro Paran	4	5
Rhulad Sengar	6	6
Whiskeyjack	7	7
Anomander Rake	10	8
Kalam Mekhar	8	9
Ammanas	9	10

Degree centrality and PageRank mostly agree on the most important characters in the series, but they start to greatly diverge the further they get away from the top 10. As shown in the figure to the right.



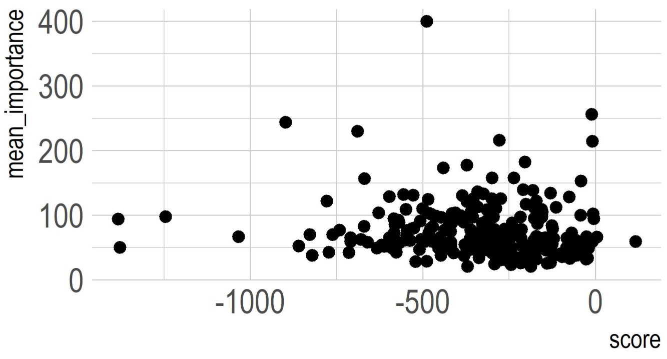
In the following network graph, I used PageRank to calculate the centrality because I don't feel that degree centrality is as suitable for this network.

The only interesting observation I can make from this graph is that there is a clear distinction between the two main continents from the



4 first books in the series and the third main continent introduced in book 5.

Before I begin trying to find the best way to find the climaxes in the series, I'll check if a chapter's sentiment score is corellated to it's importance, since I initially thought that either of them could be used to predict climaxes on their own. The importance of the chapters was calculated by taking the mean of the edge betweenness centrality values of all of the edges in each chapter.



It appears that there is little to no correllation between a chapter's mean sentiment and mean edge betweenness centrality.

Now, I'm going to try to see whether or not I can find the climaxes of the series using the mean edge betweenness centrality of each chapter weighted with the total sentiment score of each chapter.

Book	Chapter	Mean importance	Sentiment score	Weighted importance
1	2	65.01587	-319	-20740.06
2	7	89.44080	-382	-34166.38
3	7	91.34737	-508	-46404.46
4	2	122.10765	-779	-95121.86
5	25	70.48937	-762	-53712.90
6	7	50.43508	-1378	-69499.54
7	9	129.31839	-597	-77203.08
8	22	230.12420	-689	-158555.57
9	15	243.86157	-898	-218987.69
10	24	94.44786	-1383	-130621.38

2-7 Heboric skullcup dessert treck 1-2 Pale 3-7 nothing 4-2 Karsa slavery 5-25 Climax 6-7 Battle of yghatan tunnel 7-9 A lot of gods and important characters meet 8-22 Many incredibly important characters meet separately 22-24 Part of the climax 9-15 A major leader dies 10-24 Final battle of the series

While all of these chapters may be considered climaxes (with the exception of Book 3 Chapter 7), I would say that only main climaxes are for boos 5, 8, and 10. Lets look at only the mean edge betweenness centrality on its own.

Book	Chapter	Mean Importance
1	6	214.6567
2	1	153.3758
3	2	117.4767
4	5	182.7785
5	22	139.7704
6	13	110.3745
7	13	131.0852
8	22	230.1242
9	13	399.9673
10	26	256.2010

1-6 Rake tries to make alliance with darhujistan 2-1 Nothing too special just has a lot of dialogue 3-2 Tons of setup and foreshadowing 4-5 Karsa meets Leoman and joins shaik 5-22 Leads to climax 6-13 Not climax 7-13 nothing special 8-22 climax 9-13 nothing too special 10-26 Many of the main characters get their endings

The only main climax here is for Book 8, but the chapters for books 5 and 8 could be considered sub-climaxes. So just plain edge betweenness centrality doesn't quite work for finding the climaxes of each book.

What about only using sentiment scores?

Book	Chapter	Sentiment score
1	4	-371
2	6	-495
3	25	-860
4	2	-779
5	25	-762
6	7	-1378
7	24	-1034
8	24	-710
9	15	-898
10	24	-1383

Well this appears to be the best metric by far for finding climaxes. A conclusion that can easily be drawn from this is that the most negative part of each book tends to be the climax in this series. Some of these chapters have major battles, revolutions, and a couple horribly morbid chapters. Books 3, 5, 7, 8, and 10 all have what I believe are the main climaxes in their respective books, and Books 2, 4, 6, and 15 are all sub-climaxes.

While simple positive and negative sentiment scores are the best metrics for finding the climax of each book, the chapters with the highest mean edge betweenness centralities are very important chapters to the series as a whole.

Book	Chapter	Mean importance
9	13	399.9673
10	26	256.2010
9	15	243.8616
8	22	230.1242
10	12	216.1710
1	6	214.6567
4	5	182.7785
10	9	177.6568
9	16	173.5266
8	19	158.1905

Appendices

Bibliography

`https://www.reddit.com/r/Malazan/comments/a1ukxk/main_series_character_pov_data/`
affinn
all malazan books
malazan wiki
reddit people who helped make alias data