Infinite Jest: An Elegant Hairball

Hunter Wapman Brian Lubars Carl Mueller

December 13, 2018

Introduction

Infinite Jest Book

Why a Network?

Questions..questions that need answering

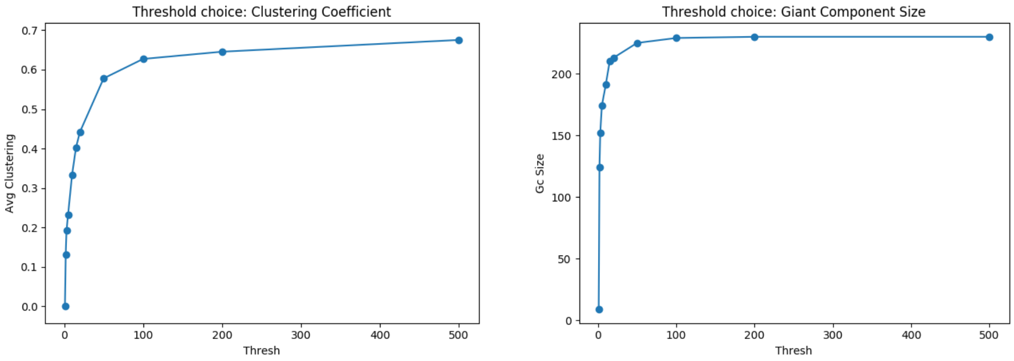
Data Processing

Data Source & Preprocessing

To generate analyzable text this project utilizes an existing '.mobi' file of Infinite Jest. This file is put through a .mobi to HTML conversion to generate an HTML formatted file as well as a .mobi to raw text conversion. This enables the use of regular expressions to find and parse various text features automatically. Thus the HTML text is used to identify endnote locations such that simplified endnote tags can be inserted into the raw text file. Similarly, regular expressions are used to indentify where section breaks exist according to the section annotations given by the book Elegant Complexity [1]. The raw text is split on these sections and saved into separate files. Likewise each endnote is saved into a separate file. In each of these files, we remove all inessential special characters (e.g. speicial quotes). As we employ the Python 3.7 standard, all text files are imported as unicode.

Named Entitiy Recognition

A major challenge with Infinite Jest is the abundant use of pseudonyms and aliases of the 200+ characters in the novel. As our network uses characters as nodes, identifying named entities and their synonym coreference resolutions (not for pronouns) requires an extensive hand-engineered approach. We utilize the Named Entity Recognition (NER) parser of the Python library SpaCy [2] augmented with its own Matcher parser. By running the existing NER model on the text, we compare candidate entities generated from the NER parser with our own indetified character matches. Manually parsing over these results enables us to build an entity-synonym list that feeds into the Matcher in subsequent NER parses. This enables Spacy to indentify a large number of the pseudonyms and aliases and their locations within each section of the text.



(a) Clustering Coefficient vs. Threshold Size

(b) Giant Component Size vs. Threshold Size

Figure 1: Comparing threshold size effect on clustering coefficient and giant component size in order to determine the ideal threshold size.

Challenges and Considerations

One major affront to the Matcher approach is the extensive use of pronoun coreferences. Similarly, the extensive use of dialog presents a challenge in identifying named entities since the the identification of the speaking character is often hidden. For example, in the following quote, two characters never explicitly mention each other but are refering to the same character, Bob.

> "*I* am really concerned about *Bob*"
> "*I* am concerned about *him* too."
> "What should *we* do about *his* problem?"

This exchange would be overlooked by our approach. While there are some approaches for coreference resolution at this granularity, most are trained on models not well suited for the very unstructured and informal style of David Foster Wallace's novel. As such, the use of named entities as nodes, and our methodlogy for identifying where they exist in text, enables the generation of a co-mention network. However, such a network may not perfectly capture true latent interaction structure of the book.

Network Design

Nodes and Edges

Network nodes constitute each character identified in the set of found named entities. These were referenced against online resources to ensure proper coverage of the characters in the book. Edges in the book represent a co-mention between two entities in the text. A threshold number of tokens (words) under which the number of tokens between the mention of one entity and another determines if an edge is established. If the edge already exists, the weight is updated. The current entity i is only matched with proceeding entities j within this threshold. Once no match is found, the next available entity is checked for proceeding matches.

The threshold number of tokens is determined by a semi-objective measure of the effect of the threshold length on the average clustering coefficient and the giant component size (see Figure 1). The intuition behind the use of these metrics is that we choose the minimum threhold length required to produce a large giant component and ample enough clustering best capturing the highly connected nature of characters in the novel. Our chosen threshold is roughly 50 tokens.

References

[1] Greg Carlisle. *Elegant complexity: a study of David Foster Wallaces Infinite jest.* Sideshow Media Group, 2007.

[2] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017.