Using Natural Language Processing to Analyse the Shape of Stories

Luca Davies
B.Sc. (Hons.) Computer Science

20th March 2020

Declaration

I certify that the material contained in this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work.
I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Name: Luca Davies
Date: 20/03/2020

Abstract

This report examines the application of natural language processing and sentiment analysis to fictional texts in an attempt to summarise narrative arcs as a curve on axes of positive/negative sentiment against time. VADER is used to process text for sentiment analysis and further experimentation is carried out to analyse how suitable VADER may be for this task. Corpora was mainly sourced from Project Gutenberg. A tool was developed to carry out a range of experiments on sentiment analysis that employed VADER as its sentiment analysis engine. Both quantitative and qualitative experimentation were carried out in the context of sentiment analysis around the following topics: the coarseness of sentiment analysis; the accuracy of VADER at sentence level via hand-tagging; sentiment analysis vs. analysis by human readers; Early Modern English vs. Modern English. It was found that Shakespearean texts proved difficult to process and so testing was also conducted using texts that had undergone spelling normalisation. It was discovered that a more powerful and versatile sentiment analysis engine would likely be needed to produce more readable curves and that a more flexible tool would prove useful for processing different text styles, such as scripts.

Working documents such as source code, full experimentation results, early planning material and a full git history may be accessed at `https://www.github.com/lucadavies/SCC300/`.

Contents

1 Introduction

1.1 Overview

Writer Kurt Vonnegut suggested at various points in his career that all stories may be categorised into a relatively small number of basic archetypes based upon the emotional ups and downs experienced within the narrative. He gave each of these curves novel names such as Man-In-Hole and Boy-Meets-Girl as a simple reference for each. (Vonnegut 2004)

Literature is one of the defining features of the human race. No other creature on Earth can write, let alone writes about itself. Humanity takes this a step further again, making up fictitious stories, some rooted in reality, some with no basis at all, in which we dream and imagine worlds and situations that may never be possible to achieve. Through written word we express emotion. All emotions, happiness and sadness; anger and fear; love and hate; awe and grief – everything humans can feel, we write about. We create characters to express and receive these emotions; they act as vehicles to transfer their feelings to a reader. With all this freedom to write and create without bound, are we actually as free as we perceive?

Vonnegut suggests that all stories are members of one of a very small number of categories of story that define roughly how the emotional progression of that story pans out. Do writers naturally and unknowingly write literature that falls into these types or is each story as unique as the next, taking its readers along its own path as they go?

This concept and these questions drew me toward this project proposal. Not only is it a fascinating endeavour to map the emotions of a novel, but as an exploration of the freedom of writers to convey emotions and of how humans often generate categories and groups without even trying.

With the high-level view clarified, it's important to note the types of tools within natural language processing that are relevant, namely that in this study, sentiment analysis takes the spotlight. Emotional analysis is a more in-depth field that shifts focus toward the psychological analysis (Anwar 2016) and employs machine learning and artificial intelligence to further predict and understand the emotions presented in text.

From a figures perspective, the basic process driving this project is as follows: a text is handed to a sentiment analysis tool which takes each sentence (or other identifiable section), processes it to produce a numeric value corresponding to that section's sentiment, then plots that value on an X-Y plot of sentiment values against progression through the text (e.g. by percentage along its course). This is the basic premise of how the processing of corpora takes place. This is discussed in more detail in later sections.

1.2 Motivation

At its core, this study is novel. It is interest driven - to attempt to show that stories can be categorised in a very simple and easy manner based upon the emotional arcs they lead readers across is an interesting new way to sort fiction. There may not be any explicit *need* to categorise literature in this way, however it has the potential to lead to further studies that examine just how it is that humans create literature and the patterns we may and may not follow.

As the project progressed, the goals shifted somewhat, leading to slightly differing aims, less focused on curve matching, and more on the use of natural language processing in literature as a whole. In some ways, the project was self-driven following initial results of attempting to identify curves. After it had proved to be difficult, the next steps naturally became to analyse quite why it was so difficult and to understand factors (such as the language and type of corpora used) that may be involved in the process.

1.3 Aims & Objectives

The aims of this report are as such:

- Design and develop an application to process a range of corpora to produce a graphing of the emotional arc during its literary course (as produced using SA)
- Analyse a range of corpora for compliance with Kurt Vonnegut's theories and story shapes
- Otherwise attempt to identify potential trends in the texts processed, such as obvious geometric differences between literature generally considered happy/sad

- Present graphs of texts to readers who are familiar with the text to assess if their perceptions of the text align with the SA graphs
- Assess VADER's ability to process text outside its design remit (e.g. Early Modern English)

1.4 Report Structure

The remainder of this report will discuss relevant background and context, the design and implementation of the Sentiplot tool, followed by a detailing of the experimentation carried out. The report will then be concluded by an analysis of the results acquired from this experimentation.

2 Background

2.1 Overview

This chapter will examine and summarise existing literature and studies in this area and topic. That is, natural language processing and sentiment analysis as a tool for extracting statistical data that describe emotions in (mainly) fictional works of literature.

The processes involved for finding useful information included searching for online articles and pre-existing projects using Google Scholar, Lancaster University Library OneSearch and Kurt Vonnegut's own lectures on this topic, while also searching for libraries to use in the implementation.

Existing projects will help to prove the developed application is performing up to standard (or not) and can be used as a side-by-side comparison of literature processed for this report and by these projects as well as to examine the processing of literature unavailable for the purposes of this report.

2.2 Natural Language Processing & Sentiment Analysis

Natural Language Processing (NLP) is a very broad field concerned with, at a high level, the understanding of human language by computers. It fuses linguistics with computer science to not just parse but to *understand* human language, to understand it in such a way that meaning can be deduced and emotion and sentiment can be extracted, even when not fully clear. More formally: "Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things. NLP researchers aim to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform desired tasks." (Chowdhury 2003)

Via the beginnings of machine translation, NLP has existed as a research field for decades, even before the current name was coined. Breakthrough projects like ELIZA (Weizenbaum 1966) made progress in the field, but it wasn't until the 1980s that statistical methods began to be used. Prior to this, large sets of hand-written rules governed how NLP systems worked, which, although sometimes effective, limited the overall progress to the manual effort put into the models and rules. In modern NLP toolkits, models are still prevalent, but instead of them being assembled by hand, they are often formed using machine learning techniques based upon employing training data, test data and then real language data.

Sentiment Analysis (SA) is a subfield of NLP that focuses on extracting and subsequently quantifying opinion and sentiment around a topic simply by analysing text. Accordingly, Liu (2012) describes sentiment analysis as: "the field of study that analyses people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities... and their attributes."

By contrast to NLP, SA is a somewhat newer field, beginning to appear formally around the turn of the century. Likely it is that opinion mining has existed for a longer time in some form, 'Sentiment Analysis' has only more recently started to appear in papers. Initial research concerned itself with extracting opinions from online product reviews. Over time, with social media services like Twitter becoming prevalent, researchers have started to apply SA techniques to text taken directly from these services. As in the case of the Hedonometer Project (see Section 2.4, Reagan et al. (2016)), this data was then plotted against time to analyse general human opinion as a given point in time. This linked in quite nicely to this project's mechanisms. With both NLP and SA garnering more and more interest in both the industrial and academic sectors, there are many all-in-one tools comprising both the NLP features required to perform SA but also tools for conducting SA itself. Among these are Stanford CoreNLP, a full NLP suite, implemented via a pipeline model, (Manning et al. 2014); LIWC, a system with a focus on emotion for psychological analysis, (Tausczik & Pennebaker 2010); Natural Language Tool Kit, NLTK, a definitive tool set for nearly all NLP operations, (Loper & Bird 2002); TextBlob, centred on text classification, built upon NLTK, (Loria et al. 2018); and multiple others. Many tools link between each other or are compiled from the best performing sections of other tools. Some of these tools are standalone, and do not provide all NLP functions in one package, instead trying to perform just a single function well - VADER, Valence Aware Dictionary and sEntiment Reasoner (Hutto & Gilbert 2014) is a common tool for sentiment analysis, used in NLTK and elsewhere via the many language ports that have been written. As discussed in Section 3.1.2, VADER was also the SA tool of choice for Sentiplot.

2.3 Kurt Vonnegut on "The Shape of Stories"

Kurt Vonnegut described a number of potential story types as displayed below in Figure 2. He suggested that all stories fit into a very small number of categories, and moreover, that stories from different cultures around the world may generally trend toward different story types compared to elsewhere. In his lecture on the topic Vonnegut (2004) draws out the curves of some well known novels and stories to demonstrate his meaning. Drawing distinct curves with very defined turning points and changes in direction that he claims match up with points in the given literature. This lecture was given a little later in his life, but the original concept stemmed from Vonnegut's rejected Master's thesis proposal presented to the University of Chicago. It was rejected by the University's Department of Anthropology on account of it seeming to simple and too much like fun. Consequently, it took a number of years before the topic was formally explored. Vonnegut's plans did however form the inspiration for this project as a whole, even if it has developed beyond his original ideas.

Figure 1: Vonnegut suggests graphing sentiment in stories against narrative progression through the story.

2.4 The Hedonometer Project

The Hedonometer project was established to gauge happiness, the world over, starting with Twitter and other social media outlets. The project's scope has since widened to process corpora direct from books, film scripts, news outlets and other foreign language literature. The overall premise is more vague than that of this study, aiming more generally to gauge happiness as a whole, but this naturally leads itself to measuring happiness against time, temporal or narrative. Less complex techniques are used than most NLP and SA, instead using a bag-of-words approach, for example on 50 million random Tweets to assign a given day a score according to Twitter. The same type of method was applied to various classic novels, including all the *Harry Potter* books. The results produced across all their various projects are very interesting and something this study hopes to replicate with more advanced techniques.

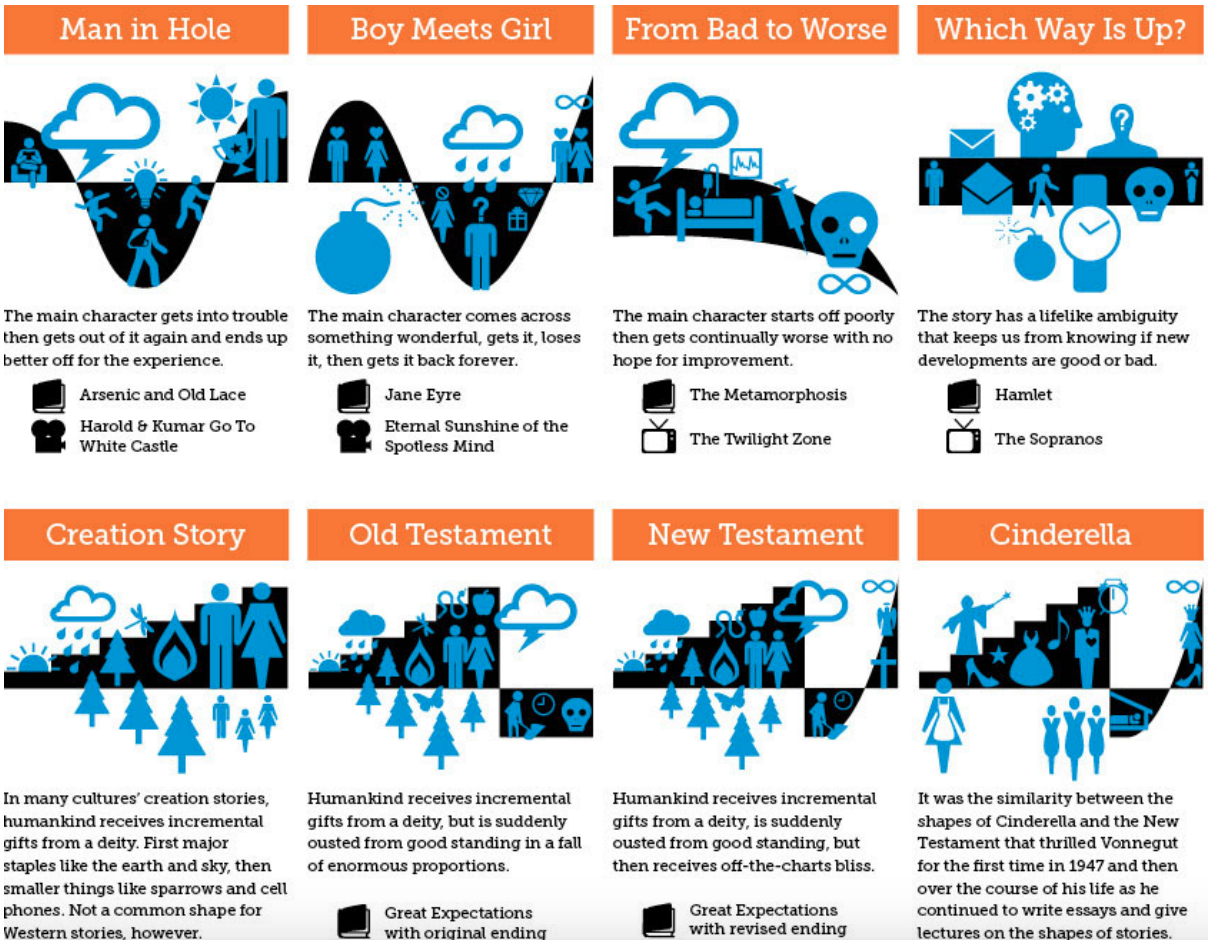


Figure 2: Vonnegut's story types. Image from a digital article discussing Vonnegut's thesis (Jones 2014), information originally sourced from two of Vonnegut's own books A Man Without A Country *(Vonnegut & Simon 2007) and* Palm Sunday *(Vonnegut 1981). Some of the Hedonometer Project's results show support for these curves.*

3 Sentiplot Tool

The experimentation required to undertake this study necessitated the need to develop a tool to process corpora and subsequently display it in a readable and analysable manner. The *Sentiplot* tool filled this role: capable of parsing text from plain text files, Project Gutenberg formatted HTML and VARDed Shakespeare XML files (see Section 4.2). This section will detail the design and implementation of *Sentiplot* during the course of the project.

3.1 Languages & Libraries

Within reason, any language could have been used to perform this study, however, ease of programmability and extendibility were also high up in the list of priorities. Familiarity with a given language and its immediately available tools and APIs held significant sway during the selection process. Together with this, the appropriate NLP and SA toolkits had to be selected to provide a good balance between usability and performance which *also* needed to be language compatible. This section briefly discusses the choice of programming language and NLP/SA tools used in the Sentiplot tool.

3.1.1 Top-level Language

When selecting which language would be most appropriate to use for this project, my major considerations were my prior knowledge of potential languages and the ease of creation of a pleasant user interface. Availability of NLP tools in each language was also to be considered, but as the end implementation shows, this was not imperative due to cross-platform and cross-language capabilities of the chosen combination of languages and tools.

A listing of considered languages follows:

- Java
  - Strong language knowledge and familiarity
  - JavaFX and Swing available for interfaces
  - Native language of Stanford CoreNLP
- Python
  - Very limited language knowledge
  - No knowledge of interface building
  - Native language of industry standard NLTK
- C# (with .NET)
  - Strongest language knowledge and familiarity
  - Extreme ease in creating interfaces via Windows Form using Visual Studio
  - Cross-platform variants of CoreNLP and VADER available via simple packages

Taking all these points into consideration C# was selected, due to its pros specific to myself and the availability of non-native libraries via APIs. This permitted me to use the full Microsoft Visual Studio suite for development, including the Windows Forms designer.

3.1.2 Natural Language Processing Tools

Various tools across a wide array of languages are freely available to provide standard NLP functions and advanced processing capabilities.

Initially a full CoreNLP pipeline was employed to process corpora but this proved to be extremely slow, loading around 2 gigabytes of models very slowly into memory before even processing anything. This is due to the various part-of-speech tagging and additional NLP tasks the pipeline had been setup to perform. These operations require usage of extensive models as reference for the application to run against. The pipeline was modified and optimised such that it only tokenised and sentence-split the input, and the actual sentiment analysis was performed by VADER.

Both the CoreNLP and VADER libraries are non-native to C# but have easy-to-use APIs for direct manipulation of their types and methods outside of their native environments. CoreNLP has an in-house developed API for C# and VADER has a third-party API called *VADERSharp*.

3.2 Design & Development

Sentiplot was developed over the course of two academic terms between early October 2019 and late January 2020. The first few weeks involved mainly research into what language was to be used and what NLP/SA tools or toolkits would provide the best mix of suitability-to-task and ease of use. The initial framework and basic features were developed using a waterfall type methodology before later, more experimental features were developed in a more ad-hoc manner using an agile development process after the main program with built. This approach was adopted based upon the time scales available: for the base application, a period of greater than a month allocated in which the bulk of programming needed to be carried out. It made sense to take this longer phase cautiously and slowly to ensure a quality code base to build features on top of in the next phase. In the feature development phase, additional smaller features were added to the Sentiplot application to facilitate the experiments (discussed in Section 5). Each of these was initially allocated a development period of up to ten days, much shorter than the previous phase, therefore it made sense to adopt a more agile, sprint-like methodology for each feature.

The base application was projected to take 4-6 weeks to develop from scratch and the smaller experimental features added after to each take around a week. In reality, this first stage of development occupied the majority of the time from the start of November through till mid December. Then each of the experimental features pushed on from there, rolling into the next term, rather than all being finished before the Christmas holidays. Thankfully this did not cause major issues as there was an excess of time set aside for write-up in the second term. Final bits of development including housekeeping and code styling wrapped up approaching week two of February 2020, still allowing time for this report write-up.

As discussed in Section 3.1.1, C# was the language used and Windows Forms was the user interface framework chosen. A full Visual Studio (Community) development suite was used to design, build and test the application which allowed for relatively swift and easy development due to the neat integration between the technologies. Had a different, less familiar language/framework been adopted, further schedule overruns may have occurred. StanfordCoreNLP was at one point the NLP tool of choice, however, difficulty and delay in marrying its Java libraries to C# and setting up its processing pipeline are in large part what encouraged the eventual swap to VADER in early December.

As the language and frameworks used lend themselves to it, the application is built with an object-oriented pattern in mind, however this doesn't come to light much as the application is somewhat self-contained - no other classes or applications need interface with it, Sentiplot only uses simple API calls to any external libraries.

The overall internal structure of Sentiplot is not massively complicated. It is detailed in Section 3.3.

3.3 Implementation

3.3.1 Structure

The application is written in C# .NET interfacing with both Java and Python libraries via APIs as detailed in sections 3.1.1 and 3.2. Windows Forms was chosen for the interface as a quick and easy to build platform, stable on Windows with seamless integration into C# and the .NET framework.

The application is composed of two forms, each with their designer code. The first allows the user to select a file to load text from and set processing granularity. The second presents the results of the analysis in multiple ways and provides facility to save these results as images of the output graphs. A high level dataflow diagram can be seen in Figure 3.

3.3.2 Sentiplot Form (Main Form)

This is the main form for the application. It is loaded on start-up and contains all needed information or otherwise calls other classes to complete its task.

Its key functions include:

- Initialising the CoreNLP pipeline to tokenise the input text
- Generating an OpenFileDialog object to allow the user to select a *.txt* *.html* or *.xml* file
- Parsing the content of the input file to prepare it for processing using regular expressions and simple character-based splitting

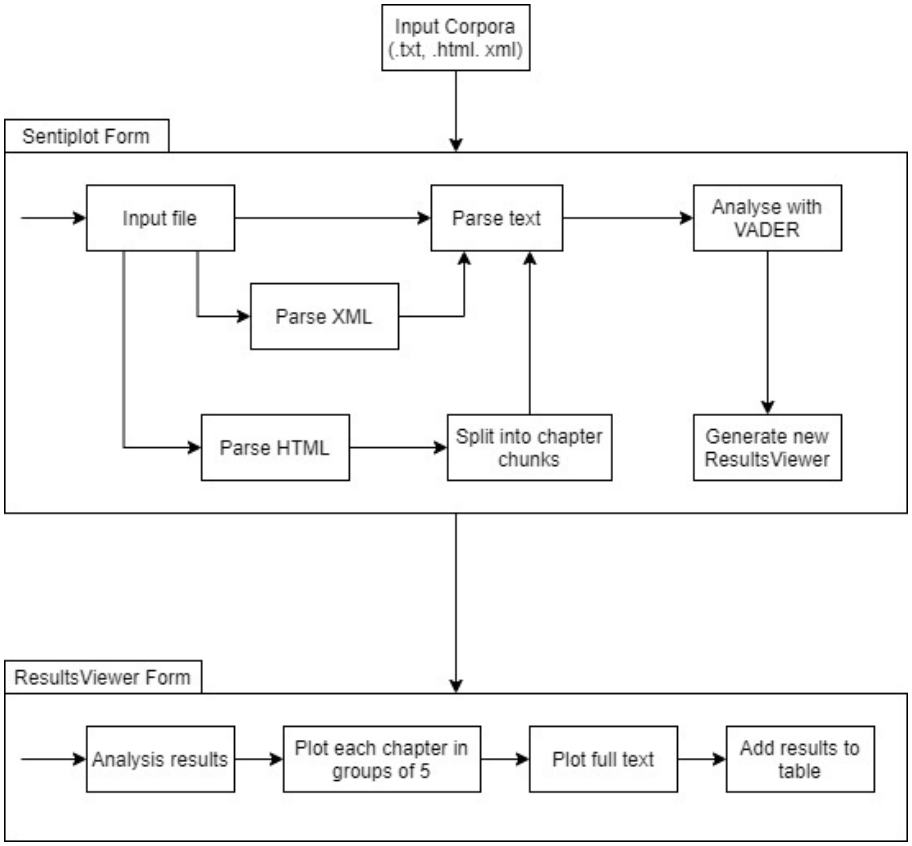


Figure 3: A high level dataflow diagram for Sentiplot.

3.3.3 ResultsViewer

This form displays the results of VADER's processing of the input text. It displays a graph of the entire text from start to finish, a full list of all tokenised sentences and their associated sentiment scores (in tabular form) and individual graphs for each chapter in the text (select HTML files only).

- Handling the output data and feeding it into the main chart and table
- Allow hiding/showing of each different type of sentiment score for the main graph
- Dynamically producing more tabs on the form to show each successive group of five chapters

3.4 User Interface

The implemented interface did not need to be excessively complex or particularly appealing as it mainly served to function as a harness to conduct the study, with more importance placed on the code-behind and results.

To provide quick results and ease of programming, the Windows Forms suite was used to produce a visually simple but functional interface. It has two screens: a screen to load the desired text to be processed and to set the granularity of the analysis, and a screen to present the processed results. The following sections provide an overview of these screens.

3.4.1 Text Selection & Options

Figure 4 shows the main presented screen, after having loaded an HTML file (*Hamlet* in this case) which has been parsed to produce plain text, having stripped out all the unneeded HTML tags.

3.4.2 Results Display

Figure 5 shows the results screen for *Hamlet*, initially showing the graph of the entire text. The maximum and minimum points are labelled with the start of the related sentence. Hovering the mouse over these

labels shows the entire sentence or sentences that produced that datapoint.

For every sentence analysed, VADER returns four sentiment scores: positive, neutral and negative (each holding a -1 to 1 score regarding match to that sentiment), and compound which acts as a single representation for the sentiment in the sentence parsed. Each of these scores have their own graph which may have their visibility toggled on and off with the check boxes in the bottom left. (Default is compound alone, as it proved the most indicative of sentiment, as advised by VADER documentation.) This graph can be saved to a JPG by clicking the save button.

Figure 6 shows the Table tab of the results screen. This simply lists each individual sentence token in the input with VADER's output value for the four scores mentioned previously. The table's default ordering is by sentence index, or the chronological order in the text, but it may be ordered by any of the fields shown.

Figure 7 shows one of the chapter tabs from the results screen. Each graphs the compound sentiment score of up to 5 chapters from the processed text. Again, maxima and minima are labelled with their relevant sentence(s), expandable by hovering the mouse over the label.

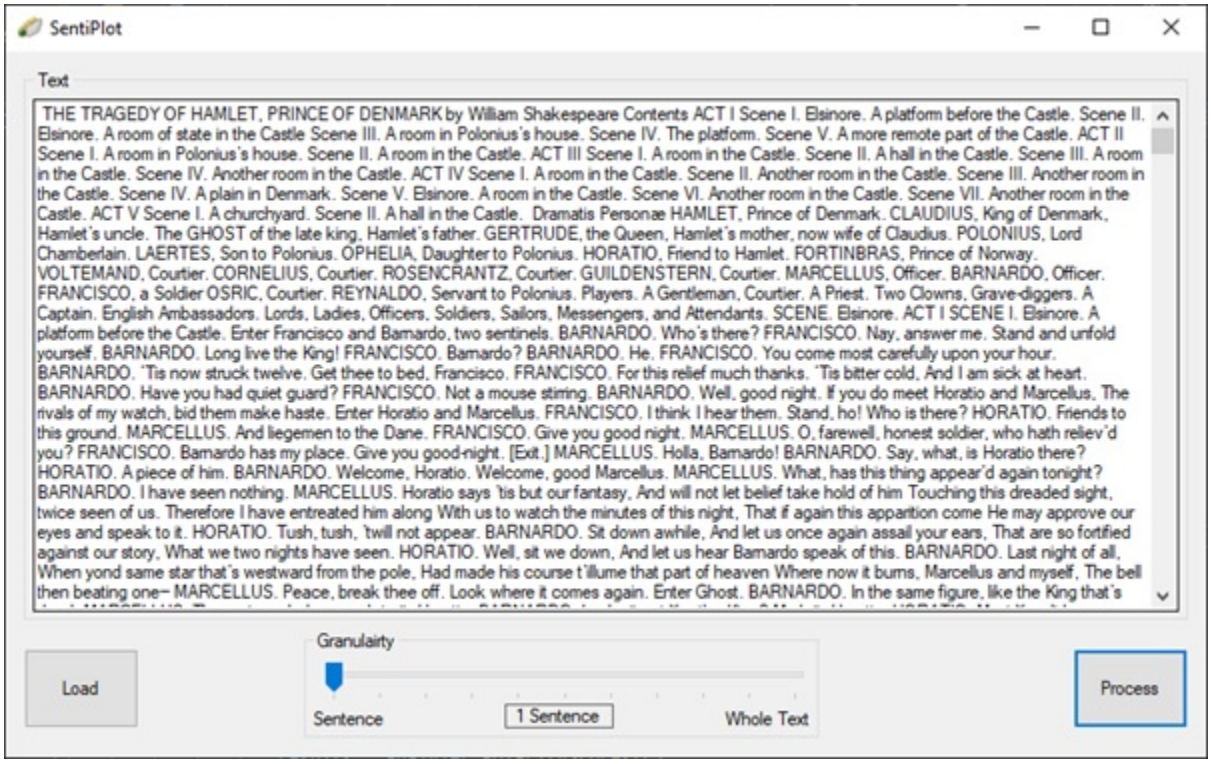


Figure 4: Main Sentiplot window

3.5 Application Summary

The Sentiplot tool was used to acquire the majority of results detailed in Section 5. Written in C#, utilising Stanford CoreNLP for basic NLP operations such as sentence splitting and tokenising, and VADER for the actual sentiment analysis. It provides functionality to process corpora raw or as provided by Project Gutenberg[1] to an end of producing a graph or curve of the sentiment prevalent in the that text, with an option to control the granularity of that processing. The same processing is also carried out on a per-chapter basis where the input text allows.

[1]For Project Gutenberg, see: `https://www.gutenberg.org/`

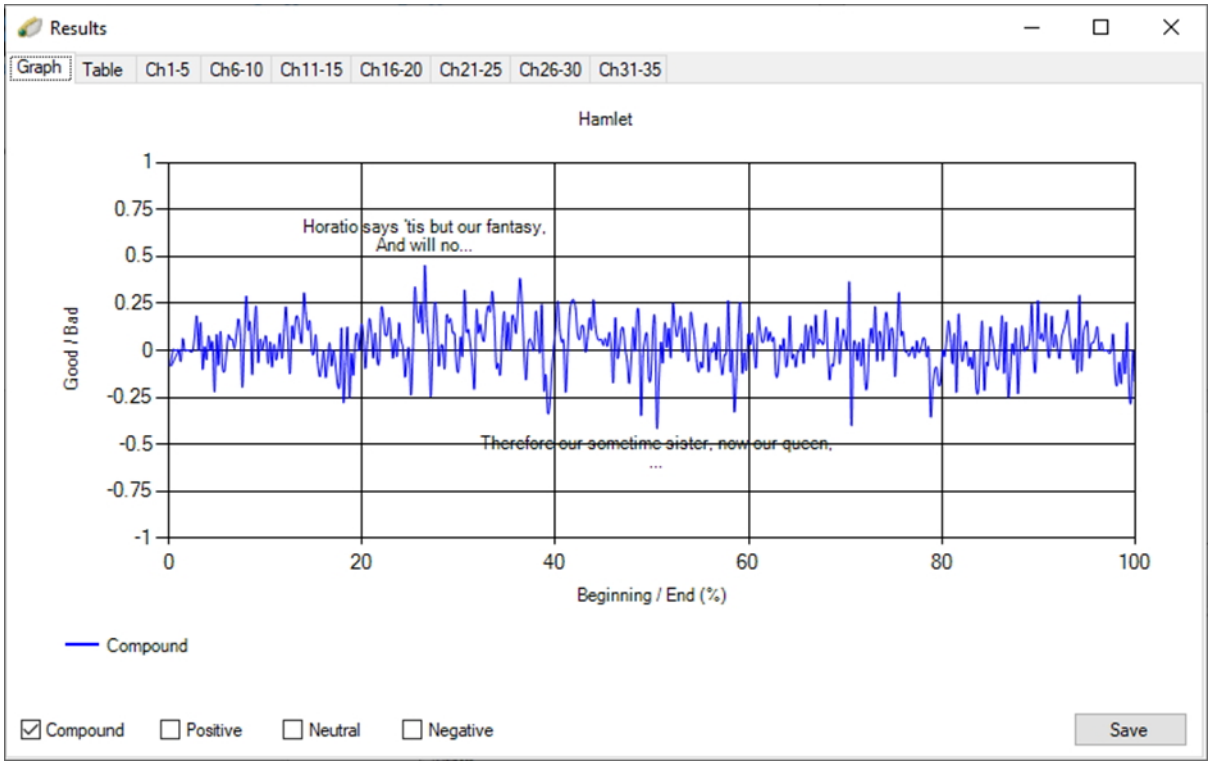
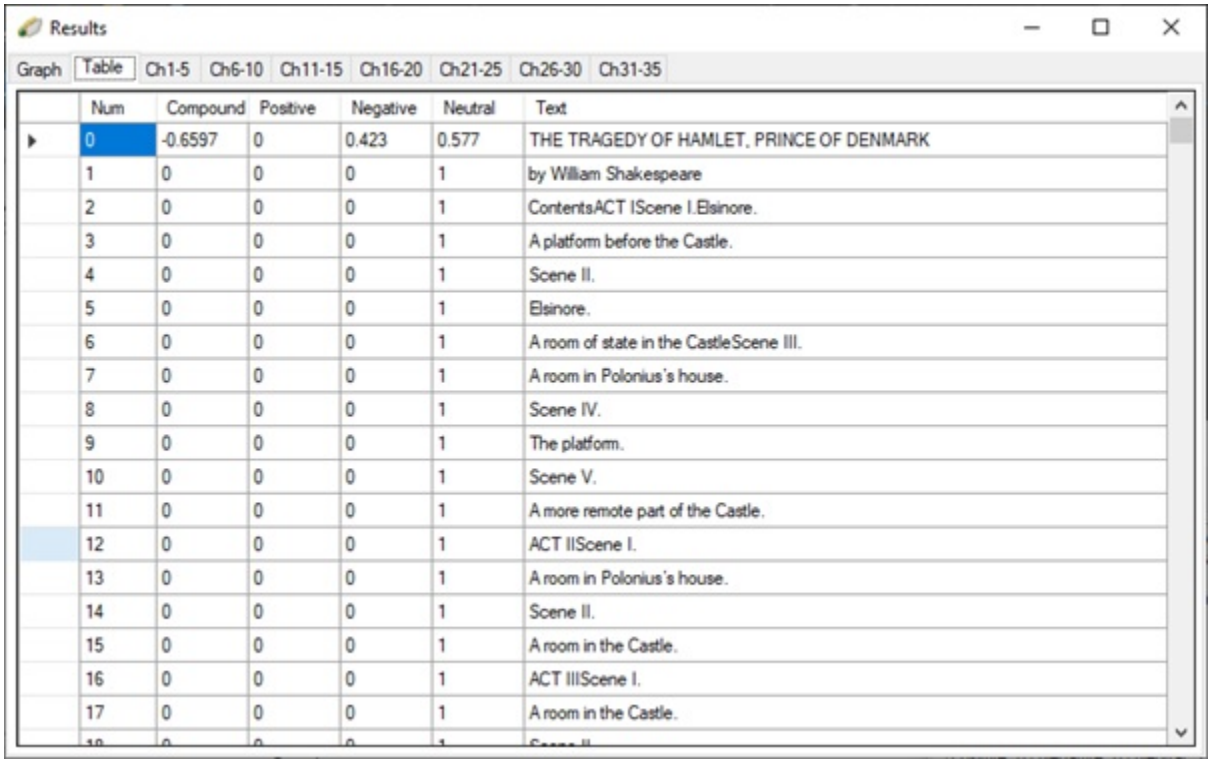


Figure 5: ResultsViewer window



Num	Compound	Positive	Negative	Neutral	Text
0	-0.6597	0	0.423	0.577	THE TRAGEDY OF HAMLET, PRINCE OF DENMARK
1	0	0	0	1	by William Shakespeare
2	0	0	0	1	ContentsACT IScene I.Elsinore.
3	0	0	0	1	A platform before the Castle.
4	0	0	0	1	Scene II.
5	0	0	0	1	Elsinore.
6	0	0	0	1	A room of state in the CastleScene III.
7	0	0	0	1	A room in Polonius's house.
8	0	0	0	1	Scene IV.
9	0	0	0	1	The platform.
10	0	0	0	1	Scene V.
11	0	0	0	1	A more remote part of the Castle.
12	0	0	0	1	ACT IIScene I.
13	0	0	0	1	A room in Polonius's house.
14	0	0	0	1	Scene II.
15	0	0	0	1	A room in the Castle.
16	0	0	0	1	ACT IIIScene I.
17	0	0	0	1	A room in the Castle.

Figure 6: ResultsViewer window showing table

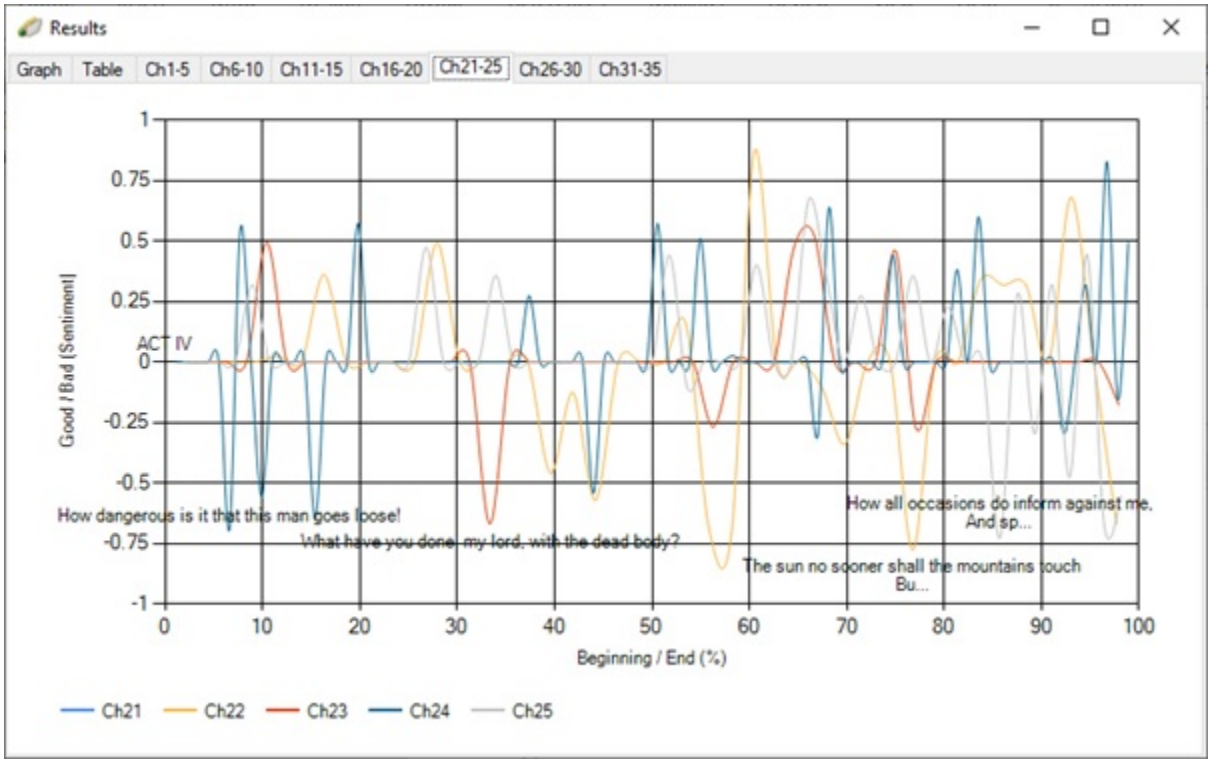


Figure 7: ResultsViewer window showing chapter graphs

4 Data

Having a tool to process a text is all well and good, but some input is of course required to produced any output. Throughout the process, considerations were being made with regard to what corpora was to be processed, graphed and examined. Much influence came from Kurt Vonnegut's lecture, mentioned in previous sections, in which he gave examples of well known stories for given curves. These included *Cinderella*, *Metamorphosis* and *Hamlet*, which were used for a majority of early testing of Sentiplot. The remainder came either from the image seen in Figure 2 or were additional Shakespeare plays.

Hamlet was the first Shakespeare processed, stemming directly from Vonnegut's lecture, but from there, processing more Shakespeare appeared to be a natural progression on account of it being so ubiquitously known. It also led nicely and easily into performing a qualitative study around perceptions of what curves of these plays could look like due to a number of personal connections with English Literature, English Language and Theatre Students with a wealth of knowledge of the texts.

4.1 Selected Corpora

The following corpora have been subjects of this report. All were extracted directly from Project Gutenberg, with the exception of the VARDed texts which were provided by Jonathan Culpeper of the Linguistics and English Language Department, Lancaster University.

- Cinderella
- Metamorphosis
- Great Expectations
- Shakespeare
  - A Midsummer Night's Dream
  - Hamlet
  - Macbeth
  - Romeo and Juliet

  - The Tempest
  - Twelfth Night

- Shakespeare (VARDed, see Section 4.2)
  - A Midsummer Night's Dream
  - Hamlet
  - Macbeth
  - Romeo and Juliet
  - The Tempest
  - Twelfth Night

4.2 Variant Detector, VARD 2

The Shakespearean texts used presented a problem unique to their age in the variety of different spellings for given words and the usage of highly unusual or archaic language. These variations can have a detrimental effect on the effectiveness of many NLP tools, particularly sentiment analysis, as unknown words may be ignored, or potentially worse, given a score opposite to what they ought to be. In order to resolve this issue, these spellings must be normalised in a process dubbed "VARDing", named for the tool used to carry out the process: the Variant Detector tool, (Baron & Rayson 2008) as produced during a study focussed on the tagging of historical corpora for use in NLP (see Rayson et al. (2007)). VARDed variants of the Shakespeare texts selected were acquired and parsed and it was these texts that were used in the Early Modern English vs. Modern English experimentation in Section 5.5.

4.3 Project Gutenberg Formatting

Chosen as the primary source of corpora, Project Gutenberg provides free, uncopyrighted literature for use in multiple formats. Early in development, Sentiplot had the capability to process plain text files only as this was deemed the simplest and easiest way to input text into the application. This was later found to be inadequate for the purposes of identifying and separately processing individual chapters.

By chance, it was found that the HTML text provided by Project Gutenberg for both *Macbeth* and *Hamlet* (two of the texts used to test Sentiplot during development) used HTML header tags to format Act and Scene numbers. It was also found that the preamble and postamble were enclosed in <pre >. . . </pre > tags exclusively in these texts. An HTML file parser was then built in to Sentiplot to strip out unneeded information (pre/postamble, HTML tags, etc), decode it, and divide into chapter sections based upon these header tags. This worked for a number of texts, however, not all. It was taken for granted and assumed that Project Gutenberg used this as a standard format for all HTML files published - unfortunately this was not the case and some of the graphs shown in this report are erroneous due to this.

Some of the texts lack any chapters that are able to be analysed and most have multiple chapters that either contain a majority of preamble/postamble or, are a single line of text or whitespace.

5 Experimentation

The following section covers the various experimentation and analysis carried out on the selected corpora. A mixture of experiment types were carried out, some quantitative, looking at how to produce the most readable output curve (see Section 5.1 Analysis Block Size), others qualitative, surveying readers of the corpora what they thought the graphs should look like (see 5.4 Reader Analysis & Reflection), and some a mixture of both qualitative and quantitative methods, such as measuring the effectiveness of VADER by hand-scoring sentences that VADER has classified and comparing (see Section 5.3, Hand Analysis Vs. VADER). The different types of experiments were carried out in order to best represent and verify any and all findings.

5.1 Analysis Block Size

Sentiplot incorporates one primary setting for its analysis of texts: the granularity slider. This allows the user to select varying degrees of granularity for the produced graphs (as a percentage of the text being analysed). This allowed analysis of texts at a number of detail levels to find which produced the most obvious curves to conform with Vonnegut's story curves. This may have introduced a small amount of positive-results bias, accepting those granularities that appear most curve-like for a given text instead of what may have actually reflected the curve of the text.

The options provided and their effectiveness are shown in Figure 8.

Too Fine	Acceptable	Too Coarse
1 Sentence	0.2%	5%
0.1%	0.5%	10%
	1%	25%
	2%	50%
		100%

Figure 8: Various granularities are available for processing text, this table shows those that proved the most effective and those that did not.

This set of experiments varied the analysis block size for a given text and compared the output graphs for the entire text with a view to identifying a specific curve for the text. The comparison is qualitative, employing only human visual reference as opposed to any mathematical function. The purpose of this part of the study was as a forerunner to further analysis and experimentation to find the best setting or the best range of settings for either specific or generically for all texts to produce a coherent sentiment curve. Figure 9 shows the differing graphs that may be produced by changing the granularity.

VADER is designed for parsing short lengths of text (less than 140 characters, as per a Tweet), so in order to produce more coarse graphs than single-sentence, VADER is still fed text on a per-sentence basis with the results summed up to N sentences where N is the corresponding number of sentences for a given granularity setting for a given text. The relevant data point is then plotted N sentences beyond the previous (i.e. at the end of the analysed block).

By default, every sentence of the text is plotted as its own point. This gives a highly detailed graph but it is very difficult to discern any sort of shape. Sentence to sentence, sentiment can vary wildly from the most negative in one to the most positive in the next (context and corpus dependent). Due to this, the resultant graph is highly spiked and difficult to garner information from.

By contrast, using the more coarse options of 25% and upward produce graphs that do show a clear line, the hope being the larger regions that have an average higher/lower sentiment score can then be picked out. However, at this level of coarseness, the data is compressed so much so that most of the detail is lost, smoothing the curve beyond useful limits for purposes of analysis

Through trial-and-error experimentation, the ideal setting was found to lie between 0.2% and 2%, erring toward the 2% for most texts. Although the choice to use percentages for the granularity setting was made to better cater to both long and short texts, those that are particularly long (>5000 sentences) and those that are particularly short (<200 sentences) may require different settings to best show the data and thus the curve. This is due to the fact that a longer novel does not necessarily have longer chapters and/or longer sections that could be identified as having a particular bias in sentiment.

With all this being said, it is possible that a different tool may have been better suited overall to this task than VADER , due, at least in part, to its native design choice to parse text the length of

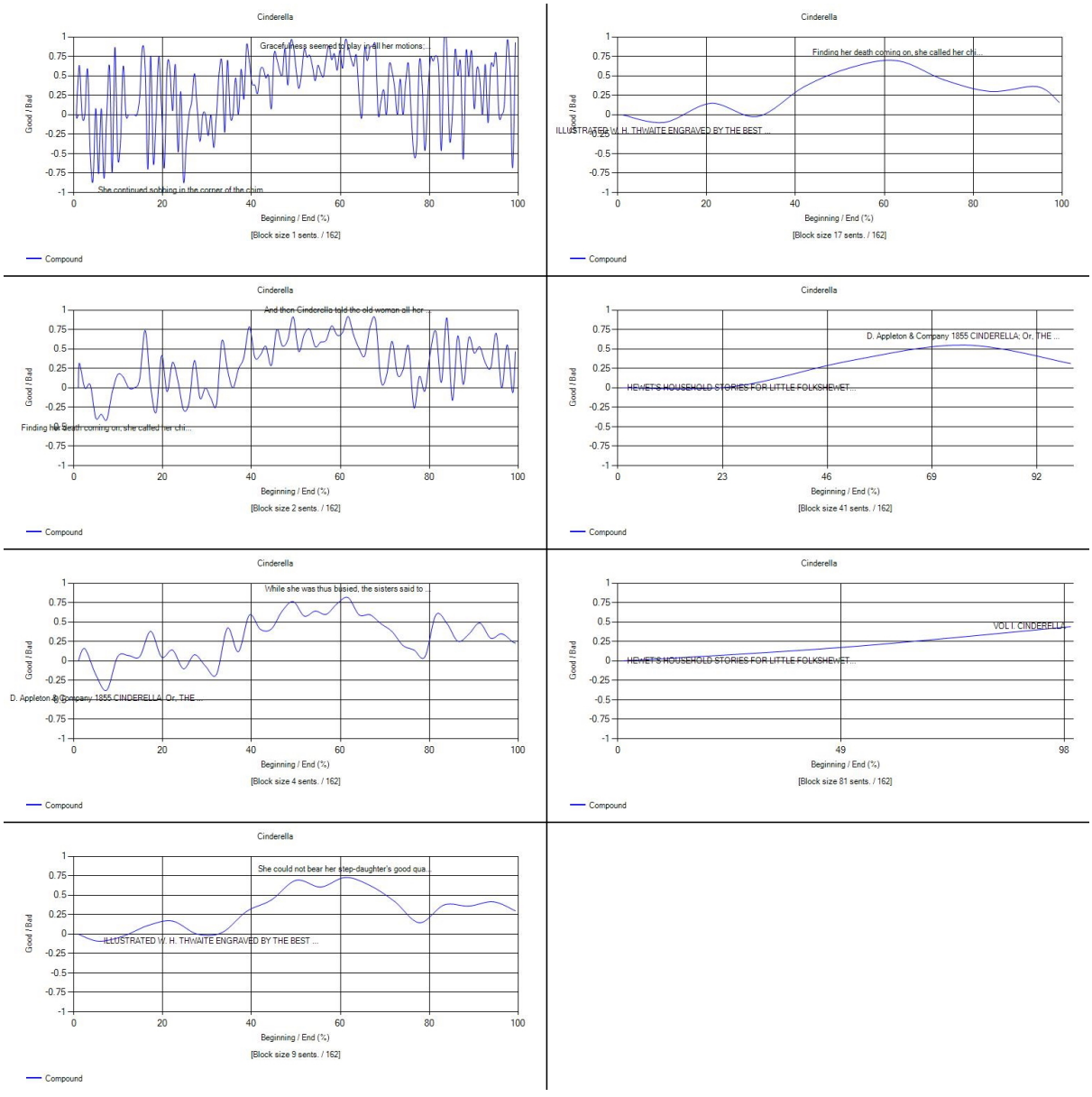


Figure 9: Each graph shows the result of decreasing the granularity of analysis and show how there appears to be a sweet spot. In this case, four of the highest options summed only a single sentence due to short length of the text. Only one of these has been included.

Tweets, not entire corpora. Sentiplot could be redesigned to pass VADER each block as a whole instead of summing the individual sentence values instead, although it is to be assumed that this would yield less accurate results as this is not how VADER was intended to be used.

5.2 Curve Identification

The original inspiration behind this project stemmed from Kurt Vonnegut's lecture on the "The Shape of Stories" and attempting to test his theory by plotting the sentiment results of various texts on a graph against time. This is what drove the initial development of Sentiplot and this study as a whole. However after initial results were produced by Sentiplot, one of two things became somewhat clear: either Vonnegut's theories were wrong, or VADER was not up to the task at hand. Which, remains to be seen and is likely best left to further research. Nevertheless, this meant that the aims of the project began to shift more toward NLP for processing literature in general (as opposed to its normal domain of opinion mining) and analysing those results. Consequently potential developments for curve identification were not developed to completion, namely, an algorithm to statistically compare the produced graphs with those predicted by Vonnegut, was never written. Despite this, mathematical plots were extracted from the six distinct curves mentioned in Figure 2 (excluding 'Which Way Is Up?' and 'New Testament' for the fact of being a flat line and identical to 'Cinderella', respectively) by using an online point plotting tool (see Rohatgi (2019)). The raw CSV data is provided in the additional documents to this report.

Due to the above mentioned reasons, in-depth analysis into finding curves in corpora was scaled back compared to original plans. The following sections briefly explore the potential loose identifications of curves in a number of texts processed by the Sentiplot tool.

5.2.1 Whole Text Analysis

A small variety of both novels and scripts of varying lengths were processed to produce their curves. As mentioned above, there were not any immediately obvious correlations between many of the texts' graphs and those predicted by Vonnegut.

Great Expectations, *Metamorphosis* and *Cinderella* were the three 'novels' processed in decreasing order of length. *Great Expectations* was the longest text processed with just shy of 10,000 sentences, however the only curve it could be said to fit at face value is 'Which Way Is Up?', or a flat line. There's variation, but only minor. Analysis of the curve from a larger perspective, one could argue it does *almost* fit Vonnegut's predicted 'New Testament' curve, albeit with an extremely reduced y-axis scale, and loss of all complex features of the graph. Kafka's *Metamorphosis* shows a similar situation: the prediction is 'From Bad To Worse', an ever decreasing parabola of sorts, however Sentiplot's output is at closest match, neutral, but at worst seems to have a few peaks and troughs of particularly negative *and* positive sentiment. *Cinderella* is the one graph that actually does bear somewhat of a resemblance to Vonnegut's prediction, being not too far removed from the rises and falls of his 'Cinderella' curve, if slightly less extreme. See Figure 10 for the produced graph overlaid with a representation of Vonnegut's curve transformed to fit the axes. The lack of the harsh drop at 70% may also be explained by the fact that block size averaging smooths out harsh changes such as this. A number of Shakespeare plays were also processed in an initial attempt to match one of Vonnegut's suggestions in his lecture via the 'curve' (supposedly a flat line) produced for *Hamlet*. Although there were some minor exceptions, the majority of these texts produced highly restricted and unvarying graphs. The reasons for this are discussed in more detail in Section 5.4. Needless to say, this makes it very difficult to even begin to classify any of the curves according to any of Vonnegut's types. Although unsuccessful at face-value, this prompted further experimentation in the form of Section 5.5 in which modernised variants of the six plays are processed instead of the originals.

5.2.2 Chapter Analysis

This functionality was implemented in Sentiplot to further examine not only if stories had curves at all, but to see if any such curves could be discerned within smaller narrative sections of a larger text. The method is not greatly complex, dividing a text into chapters and simply graphing and plotting each chapter individually to ease analysis.

Each chapter is plotted using the same granularity as the main text, this being a percentage of the input text length. Here, this percentage is applied to the length of the chapter itself. Depending on average chapter length this can mean that a specific granularity may have to be chosen to examine

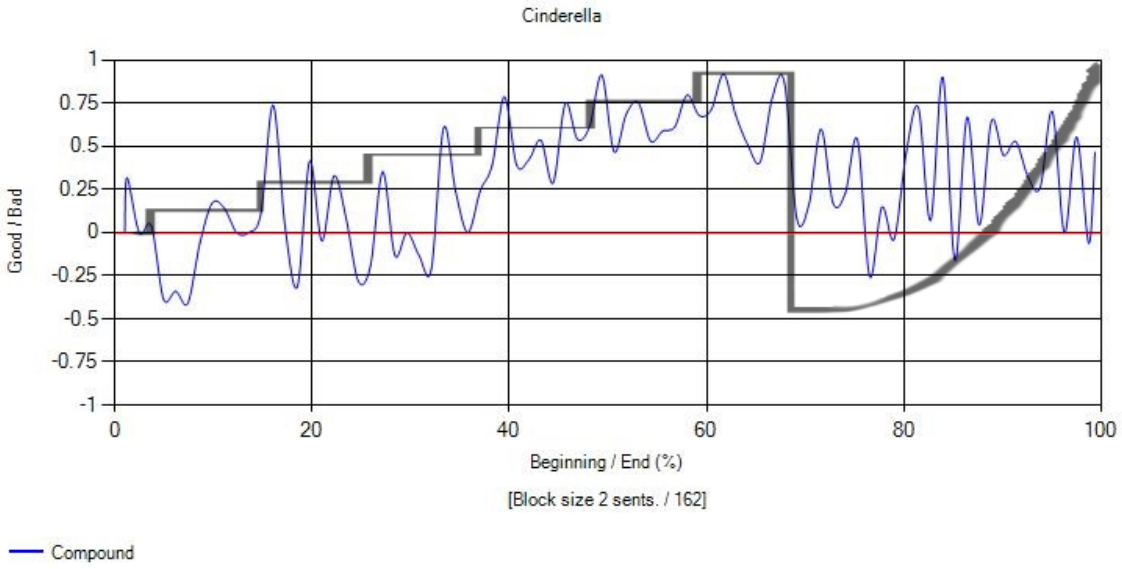


Figure 10: The curve produced for Cinderella matches somewhat closely with Vonnegut's prediction.

chapters. Being much shorter than the full text, the 1% often used for a whole text can lead to single-sentence processing for many chapters and thus yield erratic graphs. Figure 11 shows one of the chapter tabs from processing *Macbeth* with a higher percentage granularity than would normally be used, for the sake of giving useful results in the chapters screen. It can be seen here that there is a potential for curves to be found here, though given the majority negative results elsewhere, it's highly likely any matches would be either coincidence or subject to positive-result bias.

5.3 Hand Analysis Vs. VADER

This test was performed by selecting a 100 random sentences from two texts (*Macbeth* and *Cinderella*), analysing them each individually by hand and estimating a combined positive/negative sentiment score in the same range as that produced by VADER. I attempted to avoid trying to emulate the way VADER would process the text, instead scoring each sentence as I would as a reader of the text: "Would I feel positively and negatively after reading this sentence?" or, similar - a human approach.

The motivation behind this was to understand if VADER's output results were trustful, totally incorrect, or somewhere in the middle. It's easy to blindly follow what a program says but NLP and sentiment analysis is still very much a field that cannot yet directly emulate the processes of the human brain.

The results obtained, although they are based on one person's own judgement (and hence are not necessarily reliable), were disappointing to say the least. To determine similarity, a threshold difference of +/-0.25 was chosen to mark the boundary between results that correlated with each other, and results that did not. From this, of the 100 sentences taken from Cinderella, only 27% matched. In the case of Macbeth, 50% matched, but it is possible to theorise that the majority of the increase is likely due to the fact of *Macbeth* being a script - namely, there were many sentences that were just a name, or just a stage direction alone and thus was very easy to be rated neutral (or, zero) by both a human and VADER. However, in the 100 sentences, 29 names, stage directions or similar were present, of which 10 matched (mainly at or near zero), decreasing the effective percentage of matched sentences to somewhere in the range 21%-40%. There were some cases where a plain name resulted in an extreme score (see rows 4 and 9 in Figure 12). Some of the scores assigned by VADER also seem to make little to no sense whatsoever, for example, row 7 in Figure 12: the full sentence directly includes (without negation), the words "bloody", "avaricious", "false", "deceitful", "malicious", and "sin", and yet VADER scores the sentence all but neutral, while most human readers would likely rate this one of the most negative in the data set.

Though not in the previous example, it is possible that some of these erroneous results may stem from the fact that the way VADER is being utilised in Sentiplot means that it has no context for any given sentence, they are all processed stand-alone and so VADER cannot build up any running themes

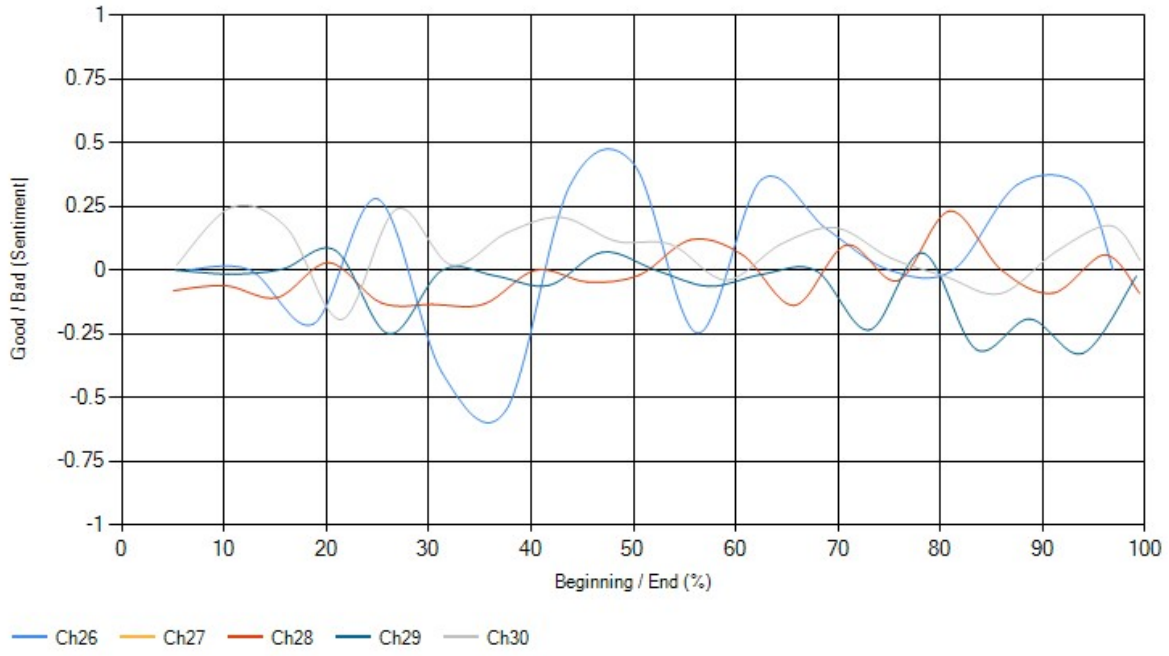


Figure 11: "Chapters" 26 to 30 of Macbeth showing five overlaid curves. Chapter 26 (light blue) shows potential shape albeit somewhat extreme.

Sentence	Human Score	VADER Score	Difference
Let grief Convert to anger; blunt not the heart, enrage it.	-0.85	-0.6093	0.24
I'll go no more: I am afraid to think what I have done; …	-0.7	-0.296	0.40
Let not your ears despise my tongue for ever, Which …	-0.2	0	0.20
MACBETH.	0	-0.6597	0.66
SCENE II.	0	0	0.00
Well, well, well.	0	0	0.00
I grant him bloody, Luxurious, avaricious, false, …	-0.9	-0.0772	0.82
or is it a fee-grief Due to some single breast?	-0.4	0	0.40
LADY MACBETH.	0	0.7096	0.71
Thou canst not say I did it.	-0.05	0	0.05
Help me hence, ho!	0	0	0.00
ROSS.	0	0	0.00
Macbeth shall sleep no more!"	-0.1	0	0.10
—A seventh!	0	0.5073	0.51
[Exit.]	0	0	0.00
List'ning their fear, I could not say "Amen," When they …	-0.3	0	0.30
I dare not speak much further: But cruel are the times, …	-0.8	0	0.80
or why Upon this blasted heath you stop our way With …	-0.25	0	0.25
Woe, alas!	-0.1	0	0.10
MACBETH.	0	0	0.00

Figure 12: An excerpt from the table of human-analysed sentiment scores along side VADER's scores with the difference (highlighted green for a match, red otherwise) shown. The original sentences from Macbeth are shown for reference.

or patterns to aid in analysis.

To conclude, this experiment raised potential issue with the usage of VADER, suggesting an alteration in Sentiplot's implementation but also questioned the overall effectiveness of VADER as a sentiment analysis tool given some rather unusual results upon close inspection. The next section partially opens up this discussion to a wider audience, allowing comment on the final results produced by VADER and some comments on its processes.

5.4 Reader Analysis & Reflection

This section covers the analysis of a range of Shakespeare plays by human readers (mostly identified to be Theatre and/or Literature students) and their personal comparison of the Sentiplot output with their own expectations of what a graph of a given play should be. The goal here was to bring a human element to the reviewing of Sentiplot and VADER. A brief background of the study, its goals, and task was given to each person who answered any of the questions provided. Some asked further questions and subsequently commented on the techniques used for analysis and graph production.

5.4.1 Reader Analysis

Readers were first asked if they felt they knew each play well enough to attempt to plot a rough arc for each on a set of axes, with a focus on any major emotional events they could identify. Some participants opted to answer questions on only a subset of the plays, as they may not have read or otherwise cannot remember the other plays. In total, 22 result sets were gathered. This allowed qualitative analysis to be carried out around how effectively VADER performed from a human perspective and to attempt to gather further comment from participants on the effectiveness of Sentiplot.

Each participant was given a blank set of axes (with the same scale as was used for VADER's results). They were asked to draw what they thought a curve of each play would look like, based upon any and all major events they can recall. They were then asked to label and/or explain any notable peaks and troughs in their curves. A small selection of the curves drawn are shown in Figure 13, paired with the corresponding output from VADER. In general, all the curves drawn by participants were wildly different to the VADER output for the same texts. As shown in Figure 14, all VADER's curves stayed mostly in the range of +/-0.25 on the Sentiment axis, whereas most participants drew curves using nearly the full extent of the y-axis. It was to be expected that participants would use the full axis, but it was not expected before setting out on this experiment that VADER's graphs would be quite so restricted. This is not to say that VADER doesn't register any sentences as reaching the more extreme values: when examining the per-sentence scores in the table view, it does, but these values get flattened when they are grouped together. On closer inspection, taking *Macbeth* as an example, approximately 30% of the sentences were outside the +/-0.25 range, however just over 60% of those *within* that range were zeros: registered by VADER as completely neutral sentences. When taking averages, these high volumes of zero-scored sentences reduced the impact on the output graph of higher scores greatly.

Even with this failure, attempting to look at relative differences along the two curve sets (those drawn, and those produced by VADER) still doesn't yield much similarity. The only immediately obvious correlation is, again, in *Macbeth* where both VADER and a participant show a sharp dip at around the halfway mark (labelled on the drawn curve as "murder"). This is nearly the only obvious match.

Kurt Vonnegut suggested in his lecture that *Hamlet* ought to have a somewhat flat curve. Sentiplot supports this at least visually, however, some of the drawn graphs show a very clear descent, logically coupling Hamlet's madness with negative sentiment. Sentiplot does in fact also show some extreme peaks and troughs, however these are momentary in the overall context of the script, and thus are averaged out and do not appear in the final output. Depending on the scope at which you examine, Sentiplot has elements of agreement with both Vonnegut's flat prediction *and* participants' more curved graphs.

5.4.2 Reader Reflection

Participants were later asked to comment on the similarity (or dissimilarity) between their own graphs and those from VADER. There was a general consensus that VADER's graphs lacked the ability to show various details and intricacies of the literature or were otherwise just plain wrong. One participant noted the failure of the system to pick up on tonal dissonances within a *A Midsummer Night's Dream* - this is assumedly a reference to the multiple plot lines within the play and their sometimes conflicting emotional

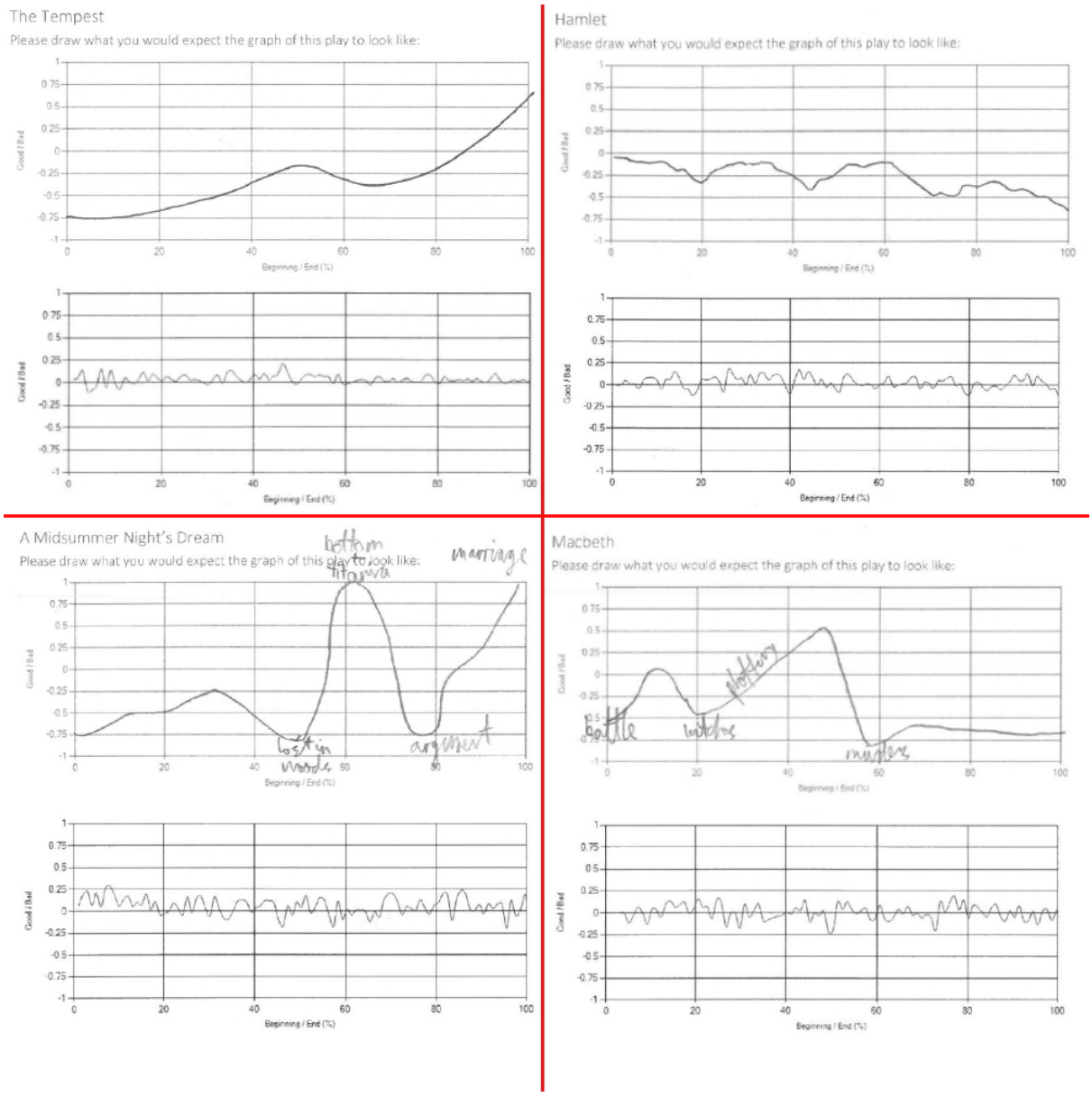


Figure 13: Readers (top of each pair) in general drew curves bearing little to no similarity to those produced by VADER (bottom in each pair)

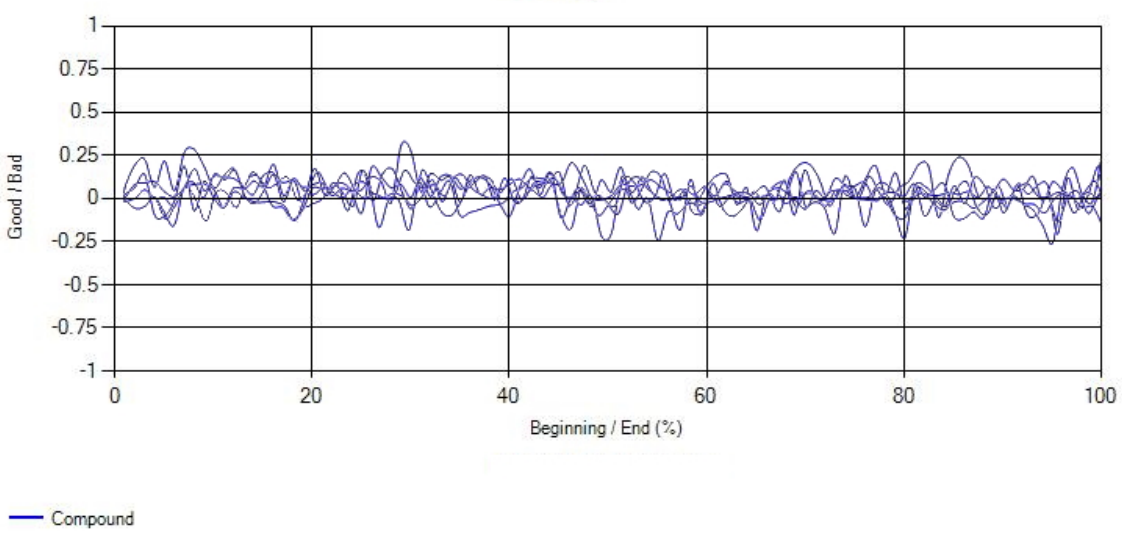


Figure 14: The graphs of six plays overlaid on top of one another. Rarely do any of the graphs exceed +/-0.25.

tones. If VADER had a greater capability to distinguish plot lines and characters within a text from one another, it's highly likely that each plot within a text would have its own curve.

It was also said that the Sentiplot graphs failed to pick up on even the extremities of character emotions (namely in *Romeo and Juliet*) and also in the overall tone of a play (*Hamlet*, showing as being relatively neutral when expected by participants to be generally negative).

Another participant mentioned that they felt undertones in the text were totally lost in the graphs (though this is perhaps to be expected) and that they failed to differentiate between specific known periods of happy/sad narrative segments. One specific example of this is seen at the end of *Romeo and Juliet* - a decidedly negative section of literature, at least at surface level. Of the two responses for *Romeo and Juliet* received, both finish the play below zero, with one plummeting the sentiment score down to maximum negative. By contrast, VADER's graph actually *rises* at the end, the second highest point in the entire graph. This is a major error, of course, but after seeing the results, the text processed was examined closer and it was found that Sentiplot had failed to strip out the sizable Project Gutenberg preamble and postamble in the processed HTML file. It was the postamble (making up the final 5% of the text) that gave rise to this upward spike where there was otherwise a negative trend that met readers expectations. Unfortunately it was found that this was not an isolated issue as already discussed in Section 4.3.

A few participants noted that some of VADER's graph appeared to have little to no correlation with the text at all, with some attributing this to a computer program's lack of the ability to experience or empathise with literary art. It was also suggested that a theatrical script is likely to be a less than ideal format for attempting to produce a curve in this way, with one participant pointing out that scripts do not contain significant amounts of description - they only contain dialogue, with a small (albeit, varying) amount of description present in the stage directions; novels on the other hand innately contain description as it is the primary information is conveyed to the reader. Sentiment analysis somewhat relies on that explicit description to score sentences, so lack of description has undoubtedly had an effect.

5.5 Early Modern Vs. Modern English

In this experiment, VARDed (see Section 4.2) variants of six Shakespearean plays were processed by Sentiplot. The VARDed texts are Modern English variants of Shakespeare's texts, the goal here being to compare just how much of an impact the difference in language (Early Modern English vs. Modern English) has on the ability of VADER to analyse sentiment.

VADER is not trained to work on the likes of formal literature. In fact, it is trained to work on highly modern and informal text, to such an extent that it can understand and analyse Internet slang and emoji. Due to this fact, this experiment was carried out to test if VADER was able perform equally or at all as well on a different form of English, that being the Early Modern English, found in the Shakespeare plays

processed for Section 5.4. The assumption is that VADER will perform better on modernised texts, as that is what its models have been trained to process.

The original Project Gutenberg texts of the plays gave graphs that did not hold much variation - processing the modernised texts gave similar graphs at first glance, but with exaggerated peaks and troughs, yet with the overall form still held. However, finer inspection will show that in fact, there's a number of new features visible in most plays that were not present in the original text. For example: in the latter 40% of the original text of *The Tempest*, the graph has less variation, staying practically in the range of zero to 0.1 - in direct opposition, the VARDed text has a number of sizable peaks and troughs, with a maximum magnitude score of approximately positive 0.3. Similarly, in *A Midsummer Night's Dream*, just after the halfway mark, there appears to be a fairly distinctive negative period for a while, a feature that isn't shown at all on the original text (in fact, it appears to become more *positive*). A selection of the results compared side-by-side with the original results may be seen in Figure 15.[2]

These new additions seen in most graphs demonstrate VADER's limited ability to process Early Modern English text, causing it produce values erring much more toward neutral zero-values than it should. This is understandable as the many unusual words used in Early Modern English will not be known to VADER and thus will likely be scored zero. It is still reasonable to attribute the restricted nature of the these original graphs, at least partially, to the nature of the text, them being scripts, not novels. However it can still be seen that processing Modern English gives more meaningful results. This is evidenced in the huge variation (nearly full -1 to 1) of the graph of *Cinderella* in Figure 9 which is also written in Modern English.

[2]It should be noted in Figure 15 that there are discrepancies between block size and text length between the original and VARDed texts. Both sets were processed at 1% granularity, however the VARDed texts lack character names in the main body of the text, as opposed to the original where they are in line. Character names were manually removed from *The Tempest* for a unique test to see if this had a significant impact on the output results. The original and the version without names produced very very similar graphs, therefore it may be assumed that the differences observed in this experiment are primarily a result of the spelling normalisation.

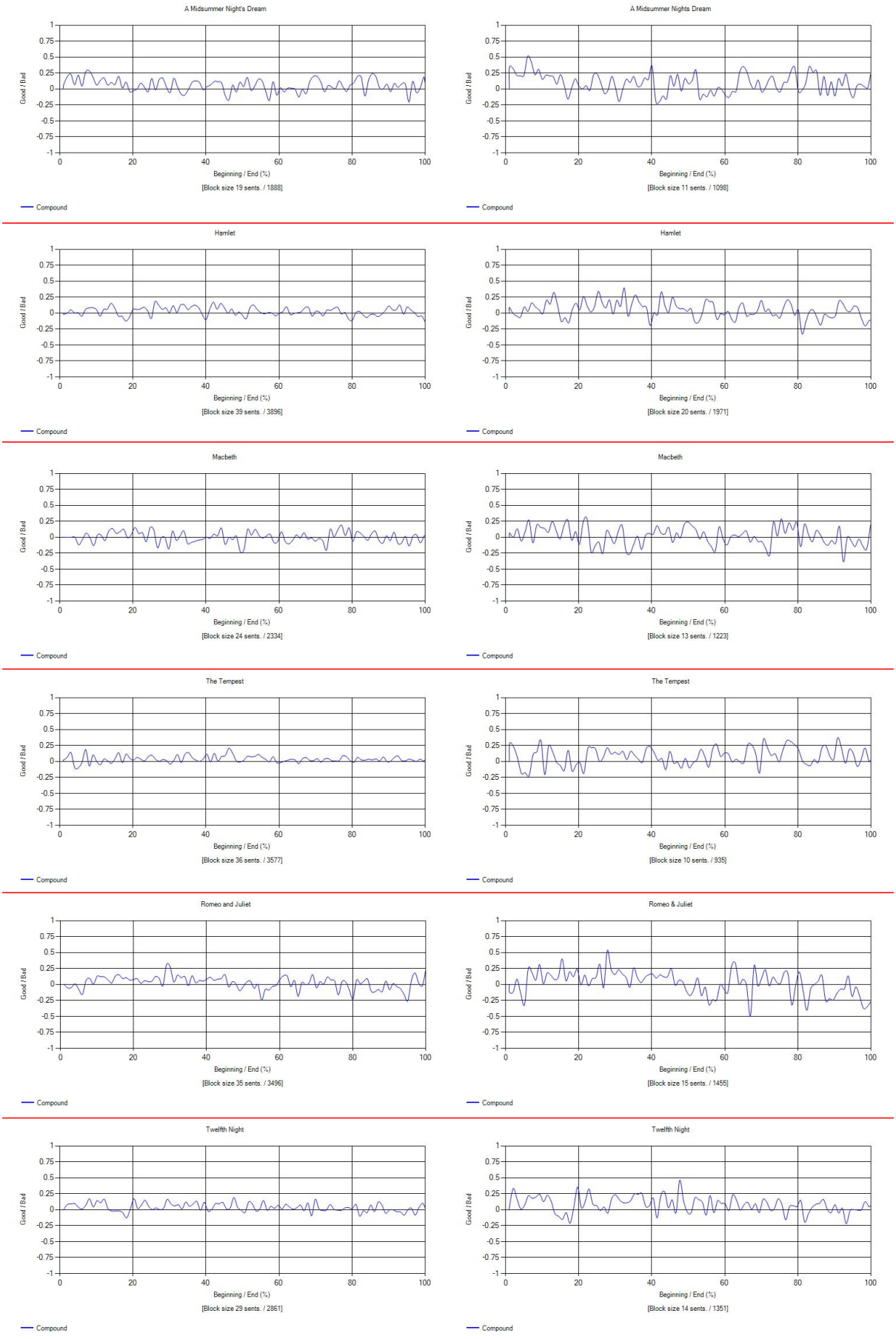


Figure 15: Original (Early Modern English) text graphed on the left, along side hand-modernised text on the right for the six VARDed texts.

6 Conclusion

This study has covered a range of topics around natural language processing and sentiment analysis in the context of fiction, both in the form of novels and play scripts. Tests were carried out to shed light on the ideal granularity of sentiment analysis in this domain and an effective range was found to be between 0.2% and 2% of the length of the corpus being processed.

Efforts were made to identify curves in literature in accordance with Kurt Vonnegut's predictions as laid out in his Master's thesis. While a clear *and* expected curve was found in *Cinderella*, the majority of the graphs produced did not show a *clear* curve, though the door remains open for further research and investigation. Likewise, obvious curves could not be identified within chapters of the same texts. That being said chapter splitting was not achievable for a majority of texts and so there is still much to be explored.

A brief analysis was conducted to attempt to assess how effective VADER has been on a per sentence basis. Sadly, results came back somewhat negative, with less than 50% of sentences falling inside the criteria to call the results a match (i.e. the difference between VADER and the hand-scoring value being less than 0.25). These results were less than impressive, yet by the time this was discovered, it was too late to swap out VADER for another tool. That being said, future extension to Sentiplot could implement this.

To further check VADER's results and to marry this to attempts to identify curves, a survey was carried out with participants who were identified primarily as students with specialist knowledge around English Literature and Language and Shakespeare through theatre. Concerned directly with Shakespeare plays, the survey comprised two sections: firstly to show what participants thought the curves of a given text would be and secondly to compare this with the curves produced by Sentiplot. Quite rightly upon looking at the presented graphs, most participants commented on how Sentiplot's graphs seemed all to look very similar to each other or otherwise bore little to no resemblance of what they thought should be shown. It was suggested that the poor variation between the graphs might be attributed to the fact that the text used to produce the graphs was a play script, which by nature contains very little (sentiment carrying) description by comparison to a novel or similar.

One of the most interesting and unique experiments was the final experiment which compared the results of Sentiplot when processing Shakespeare plays in their original Early Modern English spelling with the results of those same plays converted to Modern English by VARD. The resultant graphs showed that the VARDed texts produced much more varied results, in all case exaggerating the previously narrow-range graphs and revealing a number of additional features in a few texts.[3]

6.1 Review of Aims

The following list repeats the aims presented at the start of this report, with each aim followed by an overview of relevant results obtained.

- *Design and develop an application to process a range of corpora to produce a graphing of the emotional arc during its literary course (as produced using SA)*

  The Sentiplot tool functionally fulfilled this aim acceptably. It is able to parse text in a small number of formats, pre-process it and hand it to VADER such that the results may be plotted on a set of axes. It successfully facilitated the experimentation present in this report.

- *Analyse a range of corpora for compliance with Kurt Vonnegut's theories and story shapes*

  A small variety of corpora was processed for compliance and similarity. There was certainly scope for more texts to be analysed, however the lack of positive results and the inconsistencies of Project Gutenberg's format (see figures 5.2 & 4.3) prompted a halt to in-depth experimentation.

- *Otherwise attempt to identify potential trends in the texts processed, such as obvious geometric differences between literature generally considered happy/sad*

  Of the texts produced, *Cinderella* was the sole text to produce a visibly similar graph to to that predicted by Kurt Vonnegut. This aim was attempted and was found unsuccessful.

- *Present graphs of texts to readers who are familiar with the text to assess if their perceptions of the text align with the SA graphs*

[3] "This may be one of the first, if not *the* first comparison of sentiment analysis tools [on corpora] with and without VARD/spelling normalisation." - Paul Rayson, in personal communication

A survey of readers was carried out, obtaining 22 sets of results that was used for side-by-side analysis with VADER's graphs but also for self-analysis and comments around the Sentiplot tool.

- *Assess VADER's ability to process text outside its design remit (e.g. Early Modern English)*

  This aim was fulfilled directly via the processing of the VARDed Shakespeare texts. It was shown that VADER does indeed perform better on Modern English.

6.2 Reflections

6.2.1 Deviation From Original Plans

Originally, the main premise of this study had been to identify if, and thus to what extent, various pieces of literature could be shown to have emotional curves. These were then to be compared to a number of shapes as defined by Kurt Vonnegut. After early results came back, it was shown that this might prove to be more problematic than at first thought. Consequently, the focus of the project was widened to cover a broader remit, leading to the experiments presented in Section 5. As the project went on, each stage of progress tended to lead to the next stage quite naturally, one experiment requiring first some other information. In this way, the project was self-exploring to a small degree.

While it is unfortunate that curves could not be easily deduced from stories here, the novel origins remained present and still allowed a range of new research to be carried out in the field of NLP for fiction.

6.2.2 Revisions to System

Given the results obtained from VADER, it is likely that a different sentiment analysis engine may produce better results. This stems from a number of reasons, not least that VADER was originally designed for short input texts (the length of a Tweet) but also that it was trained on Modern English text, not the Early Modern English found in Shakespeare's plays. More advanced and versatile NLP and sentiment analysis tools may be better able to yield results on the topic

As detailed in Section 4.3, the format of corpora taken from Project Gutenberg varied in such a way that the Sentiplot tool was unable to effectively and reliably strip away unwanted text bloat before submitting it to be processed. This visually affected results at key points (the beginning and end of some graphs), causing confusion and misunderstanding amongst some of the participants in Section 5.4. An alternate, more consistent source of corpora must be found or the Sentiplot tool must be developed further to account for this inconsistency. This may mean that chapter analysis may have to be scrapped or very much non-trivial algorithms must be implemented to correctly divide texts at appropriate chapter boundaries.

6.2.3 Project Process

I conducted this project over the course of 20 weeks (plus four weeks during the Christmas holidays) from early October 2019 through till mid March 2020. The original Gantt chart (see Appendix) projected completion of the implementation by the end of the first ten weeks. While the initial design was completed in a good time, due to a range of factors outside the project remit, the development phase overran by three weeks, consequently pushing back the development of additional experiment features and further work. Fortunately, enough buffer was planned in the final write-up stage that this was easily absorbed. In future, I must to ensure my time is well balanced between academic commitments - even if a deadline is further away, a project such as this cannot be left idle so often.

I did not dedicate enough time to selecting corpora, overestimating the standardisation of Project Gutenberg which later led to some minor issues during feature development. As mentioned in Section 4.3, the formatting of the texts collected varied enough to cause noticeable discrepancies in the results due to the parser for the HTML files not being written to account for the format they were provided in. Though it had a less impactful effect on most of the results, this was problematic under the topic of chapter identification where this often produced chapters without meaningful text or failed entirely. More time needed to be dedicated to this in order for this function to work fully.

6.3 Negative Impacting Circumstances

During the final stages of the project, I was affected by two major events that took non-trivial lengths of time away from progressing my work.

First, the death and subsequent funeral of an immediate family member took place in week 18, for which I needed to return home. I was unable to work from the evening of Thursday 5th till midday Sunday 8th due to travel and lack of access to my computer and resources.

Secondly, week 20 saw the entire country's education system begin to be directly pressured by the outbreak of the Coronavirus COVID-19. Thankfully, my supervisor Paul Rayson and other involved parties were able to provide seamless assistance despite the fact I needed again to return home from Lancaster early. This did however still result in a loss of time between midday Tuesday 17th and the afternoon of Wednesday 18th, due to travel, in the run up to the deadline.

6.4 Future Research

There is scope for further research here, particularly around the comparison of NLP tool performance on different variations of English as in Section 5.5. This showed promise in the very obvious and clear improved performance when processing the VARDed text over the original. It could be explored if other tools perform better on this form of English by default, or conversely, if user-trainable tools could be taught how to process Early Modern English to an equal standard.

Sentiment analysis as a method overall may need to be tested in its own right for practicality in application to this task. There are certainly places where SA can excel: the Hedonometer Project for example, where a full cross-section of Twitter is taken and condensed into a single data point. In places where more context could prove useful, other methods may be better suited.

As a suggestion, the emotional methods used in LIWC and deeper machine learning techniques may have the potential to take the subject of NLP for fiction much further than sentiment analysis alone.

6.5 Closing Statement

The initial goal of this study may have changed significantly early on, however this report has still made fresh footprints in the region of NLP in the domain of fiction and alternative forms of English, particularly with regards to comparison of texts with and without normalised spelling. Although inferable from the underlying methods, this report goes some way to proving that normalised spelling allows NLP tools like sentiment analysis to perform better.

I have personally learnt a lot throughout the project's course, particularly in the fields of natural language processing and sentiment analysis. I have acquired new skills around project and time management, and academic writing. Alongside the great gains in knowledge for myself, I believe I have also produced a report that may provide a launchpad for other more in-depth studies in the same field.

Overall, I feel this study has been a success and that I have grown both personally and academically in light of its completion.

References

Anwar, S. (2016), 'Sentiment analysis versus emotional analysis: Same or different?'.
URL: *https://www.linkedin.com/pulse/sentiment-analysis-versus-emotional-same-different-shahbaz-anwar/*

Baron, A. & Rayson, P. (2008), Vard2: A tool for dealing with spelling variation in historical corpora, *in* 'Postgraduate conference in corpus linguistics'.

Chowdhury, G. G. (2003), 'Natural language processing', *Annual review of information science and technology* **37**(1), 51–89.

Hutto, C. J. & Gilbert, E. (2014), Vader: A parsimonious rule-based model for sentiment analysis of social media text, *in* 'Eighth international AAAI conference on weblogs and social media'.

Jones, J. (2014), 'Kurt vonnegut diagrams the shape of all stories in a master's thesis rejected by u. chicago'.
URL: *http://www.openculture.com/2014/02/kurt-vonnegut-masters-thesis-rejected-by-u-chicago.html*

Liu, B. (2012), 'Sentiment analysis and opinion mining', *Synthesis lectures on human language technologies* **5**(1), 1–167.

Loper, E. & Bird, S. (2002), 'Nltk: the natural language toolkit', *arXiv preprint cs/0205028* .

Loria, S., Keen, P., Yankovsky, R., Karesh, D., Dempsey, E., Childs, W., Schnurr, J., Qalieh, A., Ragnarsson, L., Coe, J., Calvo, A. L., Kulshrestha, N., Eslava, J., Joseph, A., Tyler, J. H., pavelmalai, Kolb, J., Ong, D. & Moschella, J. (2018), 'Textblob nlp library'.
URL: *https://textblob.readthedocs.io/en/dev/*

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J. & McClosky, D. (2014), The Stanford CoreNLP natural language processing toolkit, *in* 'Association for Computational Linguistics (ACL) System Demonstrations', pp. 55–60.
URL: *http://www.aclweb.org/anthology/P/P14/P14-5010*

Rayson, P., Archer, D. E., Baron, A., Culpeper, J. & Smith, N. (2007), Tagging the bard: Evaluating the accuracy of a modern pos tagger on early modern english corpora, *in* 'Proceedings of the Corpus Linguistics conference: CL2007'.

Reagan, A. J., Mitchell, L., Kiley, D., Danforth, C. M. & Dodds, P. S. (2016), 'The emotional arcs of stories are dominated by six basic shapes', *EPJ Data Science* **5**(1), 31.

Rohatgi, A. (2019), 'Web plot digitizer tool'.
URL: *https://automeris.io/WebPlotDigitizer/*

Tausczik, Y. R. & Pennebaker, J. W. (2010), 'The psychological meaning of words: Liwc and computerized text analysis methods', *Journal of language and social psychology* **29**(1), 24–54.

Vonnegut, K. (1981), *Palm Sunday: an autobiographical collage*, Delacorte Press.

Vonnegut, K. (2004), 'Lecture to case western reserve university'.
URL: *https://www.youtube.com/watch?v=4_RUgnC1lm8*

Vonnegut, K. & Simon, D. (2007), *A man without a country*, Seven Stories Press.

Weizenbaum, J. (1966), 'Eliza—a computer program for the study of natural language communication between man and machine', *Commun. ACM* **9**(1), 36–45.
URL: *https://doi.org/10.1145/365153.365168*

Appendix

Appendix 1: Project Proposal

Analysing the Shape of Stories using Natural Language Processing

Luca Davies

Abstract

This project proposed within is an experimental endeavour aiming to map and graph the emotional and sentimental ups and downs in fictional stories to allow further study such as comparison between different chapters or sections of the text and categorising the text in full.

The project will begin with the development of an application that may conduct the analysis. It will use NLP techniques including Sentiment Analysis and any others necessary to most effectively plot the emotional variations in the text. It will also include a visualiser to display this data on several scales from the whole text, to specific subsections. Results from the project will explore how well the application was able to identify the curve of emotions in stories and if known texts analysed produced the expected data.

1. Introduction

Stories are everywhere, fictional texts saturate the modern world narrative adverts, to films, to books. All fictional texts will include emotions: attached to a situation, a character, an object, anything essentially and these emotions are used to shape how a reader (in the case of books) feels as they read. Some stories are even specifically written to evoke particularly strong feelings in a reader toward the extreme. Writer Kurt Vonnegut suggested that all stories may follow a small number of basic story archetypes based upon how the emotions and overall sentiment in a text plays out during its course. Specifically, he proposed that one could plot on a graph, the emotions in a story against progression through the text. (I.e. x-axis as beginning to end and y-axis as positive to sad).

Vonnegut proposed seven or eight story types in common literature (detailed below). He posed the questions "*Do all stories fit into these types?*" and further *"Are there more types?*". This leads to further questions, asking if the graph or curve of all stories is meaningful in some way, or are there some which are too manic or too flat to hold any valuable information about the story.

This project will endeavour to answers Vonnegut's main questions and to go further and explore further topics. It will investigate the shape of stories as a whole before delving deeper, to finer granularity, looking at curves produced in relevance to a single character, or the graph of course of only a single chapter or section in the text and to test if the chapter itself conforms to Vonnegut's proposal of all stories fitting the aforementioned types. Time depending the project will also investigate if stories can be split into discrete sections of positivity/negativity or otherwise identifiable emotions.

2. Background

Previous projects relating to this project include Kurt Vonnegut's own *rejected* master's thesis from the University of Chicago [1]. This is where his idea of basic story archetype originated from and he has since gone on to lecture around the theory and topic [2]. Videos of his lecture can be found online [3]. Analysis has also been performed on characters and their interactions to produce a graph showing their narrative relations [4].

The Hedonometer Project [5] has essentially performed a very similar study and has graphed a significant catalogue of popular works of fiction.

3. The Proposed Project

3.1 Aims and Objectives

The aim of the project is to investigate “the shape of stories” in the sense proposed by Kurt Vonnegut. It will attempt to categorise stories according to his proposed types:

- Man In Hole
- Boy Meets Girl
- From Bad to Worse
- Which Way Is Up?
- Creation Story
- Old Testament
- New Testament
- Cinderella

The application produced will use existing NLP libraries to analyse the texts using sentiment analysis and similar techniques. It will also visualise this data in a graphical display to allow easier secondary human analysis of the data produced. Major objectives follow:

- Create an application to perform sentiment analysis on a whole text of a story – must track emotional ups and downs of the text and must visualise this data via a graph or similar interface.
- Perform case study: Investigate the shapes produced by a small number of whole texts and attempt to categorise them according to Kurt Vonnegut’s story types
- Perform case study: Investigate difference in data between small sections of the text (e.g. chapters) and the data for the whole text
- Perform case study: Investigate difference in data when it is filtered to include only those data points relating directly to a single character in the text
- Perform case study: Investigate if it is possible to identify discrete sections of the text strongly characterised by an emotion.
- The effectiveness of the NLP techniques and the results of these case studies will be discussed in the final paper writeup.

3.2 Methodology

The first steps will be to design and begin build an application that will perform the analysis on the text, tracking the emotional ups and downs to allow it to plot and graph the data at the end to gain a curve to associate with the analysed text. This part of the development will adopt a Waterfall-like model. This first stage will continue until such a time that the application can process an entire text to produce a simple graph

Beyond this, each case study will require further design/development work to expand upon the base program to allow the case studies (detailed in section 3.1) to be undertaken. An Agile work cycle will be adopted at this point to allow each case study to be examined individually. Each case study will be designed, developed, refined and carried out individually before moving on to the next.

The exact language and NLP tools to be used are not currently finalised. Candidates for use include Python with NLTK, Java with Apache OpenNLP and C# with CoreNLP. This will be finalised during the early design stage of the base application.

Similarly, the exact texts for study are also yet to be chosen. Fiction will be the target (though non-fiction works may be processed given ample time constraints as a point of interest), with no particular genre of great interest, but texts that have known stories to

myself would be desirable to easily check the correct functioning of the application during testing. Project Gutenburg provides multiple formats for all available books, including plaintext which should require very little sanitation and should be somewhat easy to parse for analysis. Exact texts will be finalised during early development stages of the application.

4. Programme of Work

The project will begin mid-October 2019 and will run till March 2020. It will be split into multiple stages as follows:

- Analysis and Design – This will involve the initial stages of the Waterfall methodology and development of designs for the application using software engineering techniques and selection of the exact language, frameworks and NLP toolkits to be used. This stage will take approximately 1 week.
- Base Application Development and Testing – This will involve the latter parts of the Waterfall methodology in development of the application from the ground up using the designs from the previous stage, testing and maintaining the application beyond testing into the case studies as further modifications are made. This stage will take 4 weeks with a chance to overrun due to magnitude of stage.
- Input Data / Text Selection – This will involve selection of several (known) texts for analysis. This stage may take place anytime between commencement of the project and completion of the previous stage. This stage will take no more than 1 week combined but may be spread over longer and interlaced to take place concurrently with earlier stages.
- Investigation – This will involve use of the developed application to analyse the selected texts and attempts to categorise them into the story types as defined earlier in this document. This stage is more of a milestone and should not occupy a significant timeframe.
- Further Investigation and Case Studies:
    - Case Study 1: Whole text vs. chapters vs. sections – This will involve the development and testing of advanced features in the application to investigate small sections of text compared to the full text.
    - Case Study 2: Per character Curve Analysis – This will involve the development and testing of advanced features in the application to investigate curves produced by analysing only lines pertaining to a single character.
    - Case Study 3: Identification of Discrete Emotional Sections – This will involve the development and testing of advanced code in the application to investigate the possibility to segment a text into discrete sections based upon the primary emotions identified per section.
    - Each Case Study should take approximately 1 week, but each may run into the next in some form (e.g. Investigation itself of CS1 may run over development of code for CS2)
- Project Analysis and Evaluation – This will involve collating of all the results from each of the investigations and combining them into a single summary and discussion document. Further, reflection upon the effectiveness of the application, how well the selected texts could be classified into Vonnegut's story types and analysis of project workflow itself will feature.

A schedule for the project is shown in a Gantt chart in Figures 1 and 2 on the following pages.

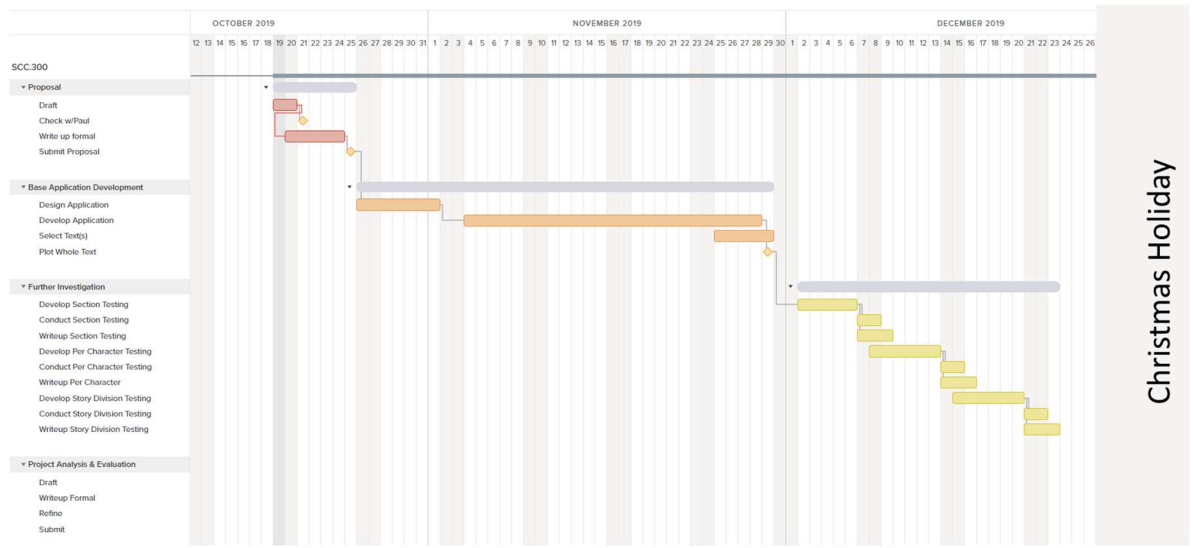


Figure 1: Project Schedule Oct-Dec 2019

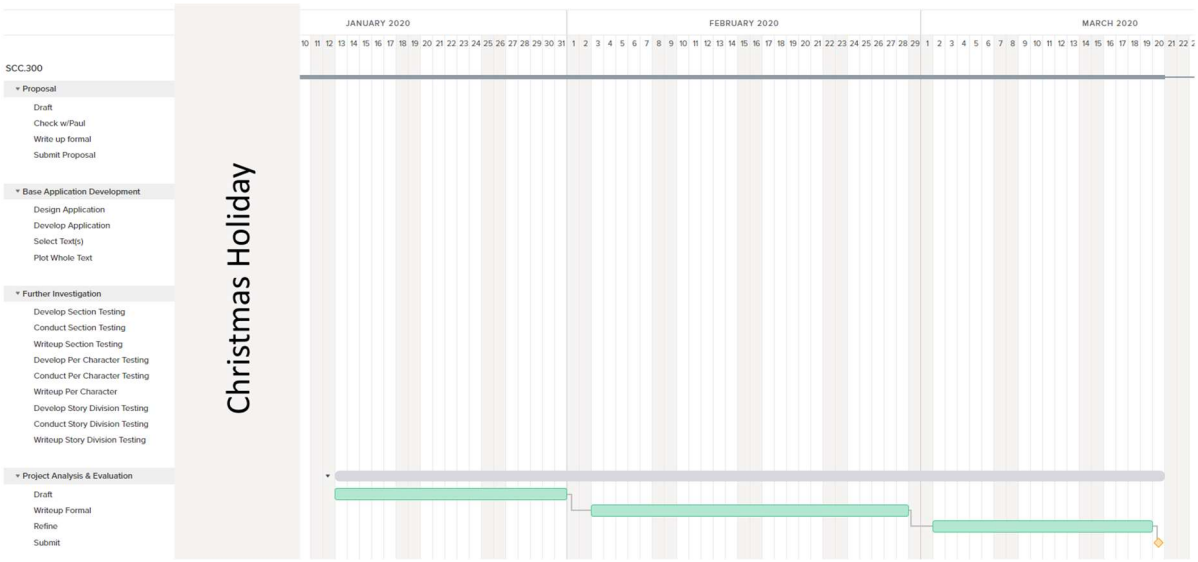


Figure 2: Project Schedule Jan-Mar 2020

5. Resources Required

Access to a software development specification computer capable of developing applications. Exact requirements for software will depend upon chosen language and toolkits.

- Python
  - Python IDE
  - NLTK and supporting libraries
- Java
  - Java IDE
  - Apache OpenNLP
- C#
  - C# IDE

- CoreNLP

Access to Project Gutenburg from which to select texts and input data.

All the above are freely available or already provisioned by Lancaster University as part-and-parcel of Undergraduate Study.

6. References

1. Kurt Vonnegut's Master's Thesis Rejected by University of Chicago
   https://whitherthebook.wordpress.com/2017/02/15/kurt-vonneguts-masters-thesis-rejected-by-university-of-chicago/
2. Kurt Vonnegut: The Shapes of Stories:
   https://fs.blog/2011/09/kurt-vonnegut-the-shapes-of-stories/
3. Kurt Vonnegut on The Shape of Stories (Short Lecture):
   https://www.youtube.com/watch?v=GOGru_4z1Vc
4. Extraction and Analysis of Fictional Character Networks: A Survey
   https://dl.acm.org/citation.cfm?doid=3362097.3344548
5. The Hedonometer Team
   http://hedonometer.org/books/v2/v2/
6. What Makes Us Happy
   https://www.turing.ac.uk/blog/what-makes-us-happy
   https://theconversation.com/what-makes-us-happy-we-analysed-200-years-of-written-text-to-find-the-answer-125252
   https://www.nature.com/articles/s41562-019-0750-z
7. Movie Story Shapes
   https://www.turing.ac.uk/blog/u-shaped-emotional-rollercoaster
   https://arxiv.org/abs/1807.02221

Background:

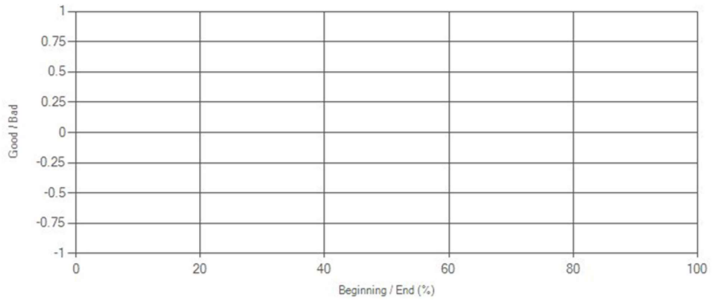
I'm using a Sentiment Analysis engine to process a number of texts (scrips, books, plays, etc...) in an attempt to try and classify them into categories as proposed by writer Kurt Vonnegut. The engine processes natural language, assigning each sentence a score based largely, but not wholly, on how positive or negative the words in the sentence are. Scores are between -1 (negative sentiment) and 1 (positive sentiment) at points throughout the course of the text. The scores for ranges of sentences are averaged to produce a smoothed graph.

As a sub-study, I am looking at the accuracy of the produced graphs when compared to people's expectations of given texts, specifically various Shakespeare plays.

Luca Davies
34653856
SCC.300 Computer Science Project:
Using Sentiment Analysis to Graph the Shape of Stories

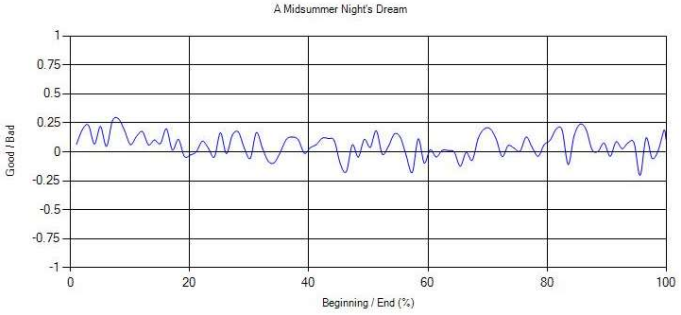
A Midsummer Night's Dream

Please draw what you would expect the graph of this play to look like:



Can you explain briefly the reasons for any peaks or troughs in your graph?

This graph produced. Can you justify differences between this and your graph?



Appendix 2: Participants Study Questionnaire