ABSTRACT

THURUVAS DINAKARAN, GOWTHAM. Visualizing Narrative Threads in a Large Collection of Documents. (Under the direction of Dr. Christopher G. Healey.)

A narrative thread can be defined as a structured sequence of interactions between independent entities. Several narrative threads can intertwine to produce plots. Visualization of narrative threads has gained importance since it can help uncover critical events within a plot. Various techniques have been proposed to visualize a plot based on character interactions, their relationships or their physical co-existence at a location. Although some techniques have considered plot sentiment, they have not been visualized at the thread-level. Unstructured text data have been growing at a rapid pace and a significant percentage is user-generated, for example, data from social media, e-mail, online product reviews, and so on. Though the definition of a narrative thread for unstructured data is unchanged in this domain, the explicit concept of plot is lost since threads are built from data, which have different origins. Instead, the data can be viewed as containing several narrative threads about a particular idea or a topic. A different set of metrics is necessary to generate and visualize narrative structures within this environment. In this thesis, I develop a novel technique to generate and visualize narrative threads from a large collection of documents. A document includes a wide range of text data, such as a customer support e-mail, a chat transcript or a comment from a social media site. I structure the documents into narrative threads based on properties derived from document clustering and sentiment analysis, then present the results as a line graph that visualizes the interaction between entities in a thread based on their sentiment.

Visualizing Narrative Threads in a Large Collection of Documents

by
Gowtham Thuruvas Dinakaran

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Computer Science

Raleigh, North Carolina

2015

APPROVED BY:

Dr. James C. Lester	Dr. Bradford Mott

Dr. Christopher G. Healey
Chair of Advisory Committee

BIOGRAPHY

Gowtham Thuruvas Dinakaran was born in Madurai, India on July 11, 1989. He received a bachelor degree in Electrical and Electronics Engineering from Anna University. He is currently enrolled in Master of Science degree in computer science at North Carolina State University, working towards fulfilling the requirement for the degree.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Christopher Healey, who guided me through my graduate education and for being a source of inspiration. I would also like to thank my committee members, Dr. James Lester and Dr. Bradford Mott, for their valuable time.

A special thanks to Kalpesh, who helped me push through various obstacles that I faced in my work and providing valuable feedback that kept me on the right track, and also to Ravi Devarajan and his team from SAS® for their valuable inputs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

A story is a chain of events in a sequence, which starts from an initial state and reaches a destination state. The set of entities (actors) in the story can interact amongst each other or with the environment, contributing to the event sequence and undergoing transformations themselves to reach the destination state. A successful story recounts a happening, whether true or fictitious, while at the same time taking the reader on an emotional ride. One basic pattern that stories follow is the hero's journey [1]. It starts with an introduction to the characters and the story environment followed by some dramatic event that breaks the normal conditions. A hero, who may be a single character or a group, becomes involved in the conflict. Finally, after an epic struggle harmony is restored. The plot within such a story is simple and predictable. However, when a story involves a large set of characters, sub-plots within the main plot, sophisticated concepts such as time-travel, or an intricate plot with non-linear narration, it becomes difficult for any reader to imagine the story in its entirety. This calls for a technique to break down and present the story based on its plot and sub-plots.

Visualization has evolved from an art form to a scientific tool for clearly, effectively, and unambiguously communicating information derived from data. This can help any user in performing analytical tasks on complex data during exploration, hypothesis generation, or decision-making. Traditionally used to analyze numerical data, the area has expanded to include unstructured input, such as text and images. Story visualization, for example,

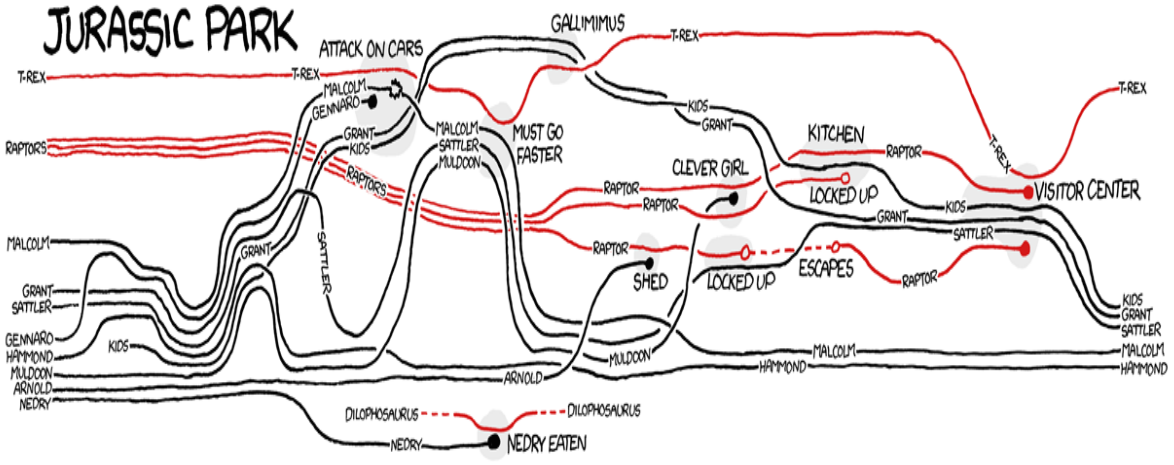

Figure 1.1 Movie Narrative Chart of *Jurassic Park* [2].

provides a new way to explore a story and helps users to unravel the plot and sub-plots within. Numerous visualization techniques have been proposed based on characters' or entities' interactions within a story, or their physical presence at various locations throughout the story. These visualizations have been useful in understanding how the entities contribute to a particular event. Thus, a sequence of interactions between independent entities can be defined as a "narrative thread," which can further combine with other threads to form plots and sub-plots. Visualization of narrative threads has gained importance since it helps in identifying significant events that were influential in determining the course of a story.

Movie Narrative Charts is one such visualization technique proposed by Munroe in one of his *xkcd* comics [2]. It shows the character interactions based on their locations during various events in the movie. Figure 1.1 shows a narrative chart of the movie *Jurassic Park*. The horizontal axis represents time. Each line represents a character. When more than one character is present in the same location at the same time, they are spatially grouped together.

In this thesis, I have developed a novel technique to construct and visualize narrative threads from a large collection of documents. A document is viewed as a stream of text data with entities interacting over a period of time. The challenge is that the threads may not necessarily contribute to the formation of a plot, since it is possible that they are from multiple difference sources. Instead, they are centered on a particular idea or topic. To overcome this problem, we developed a set of metrics based on document clustering and sentiment analysis to define and generate narrative structures. We implemented a line graph visualization to show the interaction between entities at the thread-level based on sentiment.

1.1 Motivation

According to the info graphic by Domo [3], 277,000 tweets are generated per minute, and 204,000,000 e-mail messages are sent per minute. Considering these are only two particular types of social media data, which can also include product reviews, blogs, and online news sources, one can realize the challenges involved for analysis and visualization of this type of data. In addition to these sources, many companies collect their own documents via channels such as call centers, online web chat logs, and feedback data.

An interesting property of the aforementioned user-generated data is that much of it is subjective and often the result of interactions between two or more participants. This suggests that the data should contain sentiment. The *Merriam-Webster* dictionary defines sentiment as “an attitude, thought, or judgment prompted by feeling.” To reiterate, a document could be a transcript of a conversation between a customer service representative

and a customer, a single comment thread attached to a particular post, an online chat session between two or more users, or any other similar conversation.

There is significant interest in analyzing and visualizing such large collections of documents to identify patterns and trends. For example, companies are interested in users' opinions towards their products so that they can deliver better results or identify areas for improvement. A better understanding of the data in terms of its sentiment can help achieve these goals. Different techniques exist for visualizing unstructured data, but few consider sentiment. Existing techniques display a sentiment metric of the documents across a timeline or some other property and do not consider the interaction between entities within the documents. They work well when a document is small, such as tweets, but do not perform as effectively when a document's size grows.

Our main motivation in developing our new visualization technique was to use sentiment as a significant property to distinguish documents. We also consider the interaction of entities within a narrative thread when we develop metrics for individual documents, since they provide a more accurate description of the document.

1.2 Goals

The main goal of my thesis is to construct and visualize narrative structures from a collection of documents. The narrative structures are generated based on sentiment, topic, and other relevant properties.

1.2.1 Sentiment Estimation

To perform sentiment analysis, we use SAS® Sentiment Analysis tool, part of the SAS® Enterprise Miner suite. SAS® Sentiment Analysis (SA) requires users to provide a subset of pre-classified data to form a training set. To overcome the need for manual training set construction, we propose a new method that uses sentiment term dictionaries to automate this process, significantly reducing the time required to construct training sets, and expanding the range of domains where sentiment analysis can be applied. The resulting training sets still require some post-processing to identify and correctly classify domain-specific terms, but the time required to validate a training set is minor compared to the overall effort required to generate it.

1.2.2 Document Clustering

We perform topic clustering on a document collection to extract a set of topics. Documents about a particular topic are grouped together. We use SAS® Text Miner for topic clustering since it is one of the best performing tools currently available.

1.2.3 Document-Level Sentiment

Sentiment analysis is performed on individual documents to develop sentiment estimates. Processing an entire document in one step only works well for small-sized documents, such as tweets. When the size of a document increases, simple dictionary-based sentiment analysis on the entire document yields misleading results since it does not properly identify changes in sentiment as events unfold throughout the document.

A sentiment estimate on an entire document combines even-level sentiments, which often cancel one another to produce a neutral overall sentiment. Moreover, identifying where

sentiment boundaries occur within a document is an important property to locate and visualize. To achieve these goals, each document is decomposed into a sequence of events based on the entities present in the document. A single event is usually a comment or a statement made by a particular user. We perform sentiment analysis on each event, for example, a single sentence or small group of sentences, using SAS® SA trained on our automatically generated training sets. Input to the SAS® tool is classified as positive, neutral, or negative.

1.2.4 Visualizing Narrative Structures

We structure each document in the collection into a narrative using the sentiment results obtained. We implement a line graph to visualize an individual narrative structure, which shows the entities' interactions and their corresponding sentiment. We identify critical events that produce sentiment transitions: a change in the polarity (positive, negative or neutral) of the sentiment from one event to the next within a document. Entities involved in these events are tagged with one or two sentiment keywords from the text to give the user clues about what might have caused the sentiment transition.

1.3 Thesis Organization

The remainder of the thesis is structured as follows. Chapter 2 describes various techniques involved in sentiment estimation and relevant research on narrative analysis and narrative visualization. Chapter 3 provides a detailed description of the algorithm used in estimating document sentiment. Chapter 4 discusses how documents are structured into a narrative to visualize the narrative's structure and its associated properties such as sentiment,

time, and relevant keywords. Chapter 5 applies the algorithm to a collection of real-world documents to validate its performance and explore its strengths and limitations. Chapter 6 presents future work and conclusion.

Chapter 2

Background

Text mining is the process of extracting useful information from text data. It involves one or more tasks, such as part-of-speech tagging, text clustering, topic extraction, sentiment analysis, named entity recognition, and so on. The end goal is to structure text in ways that can be used to analyze and derive information. Since complex patterns are often difficult to identify algorithmically, data visualization can help to provide a better understanding of the results. This chapter presents a brief history of text analytics and common tasks involved. We then discuss the related work in sentiment analysis and narrative in the context of data visualization.

2.1 Text Analysis

One of the earliest known applications of text analysis was the Concordance of Vulgate compiled by Dominican monks during the late 13th century [4]. It contains a list of common phrases and their occurrences in the work. The late 19th century saw the rise of text analysis in stylometry. David I. Holmes [5] describes the development of statistical methods for analysis of literary style and the struggle involved in building a technique, which may be applied to all genres, languages and eras. T. C. Mendenhall [6] was one of the first to study the frequency distribution of words of different lengths in the works of Shakespeare, Marlowe and Bacon to attribute the authorship of their works. Text analysis during the pre-computing era was a laborious task, which deterred many statisticians from pursuing further

work. The advent of powerful computing machines helped research in text analysis gain traction. Advanced data storage devices enabled the effective access to large corpora of text data. With the introduction of the Internet, this access became even more simple.

It is important to understand the distinction between information retrieval and text data mining. Marti A. Hearst describes information retrieval as the process of accessing a document that a user requests and text data mining as the process of discovering new information from the document [7]. Yet, Hearst claims certain types of text analysis, such as document clustering, can yield tools that aid in the information access process. The rise of web-based applications has given people new ways to create, access, or share information. Hearst emphasizes the use of online text collections in discovering new facts and trends. She suggests that computationally driven text analysis with human-guided decision-making alone may provide exciting results, as opposed to a fully automated text analysis approach. Many text analysis techniques follow the fully automated method, for example, by manually classifying a subset of documents and using them to train statistical or machine learning models.

2.2 Text Analysis Tasks

The major challenge involved in text data mining arises from the fact that natural language is complex and not free from ambiguity. It is possible that one word may have multiple meanings, and a sentence can be interpreted in different ways. Text data is not well organized and is normally considered unstructured or semi-structured. Statistical or machine

learning models may require human-annotated training sets to learn from. This is a manual and time-consuming process.

The rise of unstructured text data presents two major problems: (1) a way to organize a collection of documents, and (2) a way to extract information that is characteristic of a particular document. Most applications in text analysis try to solve either or both of these problems or analyze the results from these problems to derive further information.

2.2.1 Text Processing

It is necessary to process plain text documents into data structures, which are more suitable for advanced analysis. Andreas, Andreas, and Gerhard describe the common text pre-processing and document representation models used in text mining [8]. Several methods exist based on the idea that a text document is described by the set of words it contains. We use this "bag-of-words" approach in this thesis. It is the most commonly used technique in text analysis due to its simplicity. It ignores the order of occurrence and the semantics of the words within a document, but retains the frequency information of the words, which is essential in text analysis tasks such as document clustering. There are techniques to overcome these challenges to some extent, such as *n*-gram models and Latent Semantic Indexing, which are based on the bag-of-words model. Semantic understanding of text is still a major challenge in natural language processing. There exists no other successful approach in practice.

The main objective of text pre-processing is to break down each document into a set of words and reduce the size of the set by eliminating words or merging related words. Tokenization is the process of splitting a document into a set of words by removing

punctuation, whitespace, and other non-text characters. The following methods are used to further reduce a document's size.

1. *Filtering:* Filtering is the process of removing words that contribute little or no information. They include articles, conjunctions, and prepositions, collectively called stop words. In some cases, words that occur in high frequency or very rarely also convey no useful information and are removed.
2. *Stemming:* Stemming is the process of merging group of words that share the same root form. The word stem replaces the original word. For example, the words running, runs, and ran all use the stem word run. Porter proposed an algorithm for automatic suffix striping to reduce words down to their root form [9]. The algorithm makes use of an explicit suffix list. A criterion is specified along with each suffix to determine if the suffix can be removed from a word to reduce it to its word stem. The advantage of this algorithm is that it is small, fast, and simple. It does not handle the ambiguity arising from the context of a word's usage, however, such as the words "relate" and "relativity", which are both reduced to a single word stem "relat." But, Porter argues that when suffixes are being removed for improving information retrieval performance, it is not necessary to consider these linguistic challenges. He proves that the simplicity of the algorithm does not hinder its performance on the real-world data by comparing it to a much more elaborate algorithm, which was used in information retrieval (The Development of a Fast Conflation Algorithm For English [10]). The results showed that performance of both algorithms were not significantly different.

Porter's algorithm works by representing a word in terms of the vowels and consonants present within it. A consonant is denoted by `c`, a vowel by `v`. When consonants or vowels occur in series, a single letter replaces them. For example, the word "traditional" has the pattern `cvcvcvcvc`. This is further reduced to the form `[c]` $(vc)^m$ `[v]`, where `[c]` and `[v]` represent an optional occurrence of consonants and vowels, and $(vc)^m$ denotes that the pattern `vc` occurs *m* times, *m* is called "the measure of any word or word part" [9]. The *m* value is often used in a condition to determine if a rule to reduce a suffix in a word is applied. Rules are of the form

`(condition) S1 ➔ S2`

If a word ends with the suffix S1, and the condition is satisfied, S1 is replaced by S2. The condition checks for patterns, such as if the stem ends with a specific letter, or if the stem ends with two consonants, and so on. There can be more than one condition combined together by means of *and*, *or,* and *not* expressions. When a word matches more than one rule, the rule with "longest matching S1" for the word is followed. The algorithm follows a five-step approach with different rules in each step. It removes redundant suffixes, such as ones denoting plural form and verb tenses, in the initial steps. Successive steps take care of special cases and suffix groups. An example of a Porter rule is:

`(m > 0) ATIONAL ➔ ATE`

The condition states that any word that ends in ATIONAL and contains at least one `(vc)` pair can be reduced by replacing ATIONAL by ATE. For example, the word "relational" is represented by the form $c(vc)^4$ and ends in ATIONAL. Since $m > 0$, the

rule is applied reducing “relational” to “relate.” Porter’s algorithm is the most commonly used stemming algorithm for English language and has shown to be empirically effective [11].

3. *Lemmatization:* Lemmatization is a more sophisticated version of stemming that tries to return the base form of a word, known as a lemma, using vocabulary and morphological analysis. For example, “am,” “are,” and “is” lemmatizes to “be,” something stemming could not perform. Lemmatization does a dictionary lookup to try to identify the context and part-of-speech of a word. This requires tagging the part-of-speech of every word in the document, which can be time-consuming and error-prone. Stemming does not consider the meaning or context of a word and hence, is simple and fast. Hence, in many cases stemming alone is considered sufficient.

After pre-processing, it is necessary to convert the word sets to a data structure, which aids in analysis. A well-known technique is the bag-of-words model, which uses a set of words and word frequencies of occurrence as its feature. Salton, Wong, and Yang in 1975 proposed a vector space model that uses term vectors to represent a collection of documents [12]. Originally designed for indexing and information retrieval, it is now used in several text mining techniques. Each document is described by an n-dimensional vector, where n corresponds to the number of terms in the collection. A vector has a non-zero value, called the term weight, in its column if the term is present in the vector’s document.

The n-gram model uses contiguous sequences of n words from the document to form, for example, bigrams, with $n = 2$ contiguous terms, trigrams with $n = 3$ terms, and so on. n-grams are used to identify co-occurrences of words in a document, particularly when order of

occurrence is important. Latent Semantic Indexing (LSI) is another modeling technique that uses a vector based representation of documents to convey their semantic content. It is mainly used to measure the similarity between document pairs. LSI involves two major steps [13]. The first is to construct a document-term matrix from a large text corpus. The documents can be paragraphs within a single large text document or different text documents. Each row corresponds to unique terms within the corpus. The columns represent the frequency or term weight of that particular row term within a document. Next, Singular Value Decomposition (SVD) is applied to the matrix to decrease its dimensionality. This reduces the number of rows within the matrix while preserving the similarity between the columns. Similarity between two documents is calculated by the dot product (the cosine of the angle) between the documents' term vectors. LSI can be used for information retrieval and document clustering. LSI is also used in identifying relationships between words such as synonymy, where two words convey the same meaning, and polysemy, where the same word occurs in different contexts.

There are many ways to compute the term weight for a term within a text corpus. The weighting scheme based on term frequency inverse-document frequency (TF-IDF) is probably most well-known. It measures the importance of a term in a particular document based on the term's ability to distinguish the document from other documents in the collection. TF-IDF is directly proportional to the frequency of a term's occurrence in a document and inversely proportional to the frequency of the term's occurrence in the entire document collection, calculated as the product of term frequency (TF) and inverse document frequency (IDF). There are several variants in the calculation of TF-IDF. Term frequency, in

its simplest form, is the number of times a term occurs within a document. Raw frequency may be avoided to reduce the effect of a large frequency of a term within a document. Instead, logarithmic scale can be used.

$$\text{TF} = log_{10}(1 + tf_{t,d}),$$

where $tf_{t,d}$ is the raw frequency of a term *t* within a document *d*.

Inverse document frequency is the ratio of the total number of documents to the number of documents that contain a particular term. It increases when a term is present in only a small group of documents.

$$\text{IDF} = log_{10}\left({}^{N}/_{d_t}\right),$$

where *N* is the total number of documents in the document collection, and d_t is the number of documents that contain the term *t*. TF-IDF is then calculated as the product of TF and IDF. For example, consider a word *fire* that occurs four times in a document. The TF would then be 0.698. If the document is in a collection of 1000 documents, and the term *fire* is present in 100 documents, then IDF value would be 1. The term weight for *fire* for this particular document will be |TF * IDF| = 0.698.

In addition to the frequency of occurrence of words, linguistic pre-processing can be employed to add additional information about the words in a document [8]. The following are commonly applied methods:

1. *Part-of-speech tagging:* Each word has its part-of-speech information tagged along with its frequency.

2. *Parsing:* Each sentence is parsed to produce a parse tree that describes the relation between words within a sentence.
3. *Text Chunking:* This approach groups adjacent words in a sentence into phrases. It is particularly useful during the information extraction process.
4. *Word Sense Disambiguation:* Instead of storing the actual word, the meanings of the words are stored in the document representation. This is useful when semantic information is required.

2.2.2 Document Organization

One of the main problems with large document collections is in organizing them and simplifying access to target documents. There are two common approaches to solve this problem.

1. *Classification:* Document classification is the task of assigning a document to a particular class or category based on its contents. The challenge is to develop a classifier that automatically classifies the documents without manual intervention. Some of the early works of Maron [14], and Borko and Bernick [15] attempt to prove that automatic document classification is possible. Several techniques have been proposed since then. Commonly used techniques include probabilistic Bayesian models, decision trees, and support vector machines. Classification usually requires a set of manually labeled documents that will be used as training set for the statistical or machine learning model.
2. *Clustering:* Clustering is the process of automatically grouping documents with similar content. The aim is to obtain clusters, such that documents within a cluster are similar, and documents from different clusters are dissimilar. Grouping is done based on

similarity or dissimilarity scores for each pair of documents calculated using their document representation models. One commonly used clustering technique is hierarchical clustering [16], which obtains clusters that can be represented in a hierarchical manner. It can work either "bottom-up," where each document forms an initial cluster, and similar clusters are iteratively merged, or "top-down," where the entire collection forms a single cluster that is iteratively split to form sub-clusters. Another common clustering technique is *k*-means clustering. It is a simple algorithm that can be scaled to large datasets. Several variants of the algorithm are available.

2.2.3 Information Extraction

Text data contains information that cannot be readily understood by computers. Information extraction can be regarded as a restricted form of full natural language understanding [8]. The main task is to identify text with particular target attributes. The following are some common tasks in information extraction.

1. *Named entity recognition*: Identifying names of people, locations, or organizations.
2. *Co-reference*: Finding expressions that refer to a previously presented entity. For example, in the sentence, "Will won the lottery, and he was happy." "he" is a co-reference to "Will."
3. *Relationship extraction:* Identifying relationships between two entities. For example, in the sentence, "Mark is a student at NC State." "Mark" has a student relationship with "NC State."

2.3 Sentiment Analysis

The Internet has had a revolutionary impact on our culture and the way we interact. Instant communication has been made possible via chat applications, voice over Internet protocols, video chatting services, and so on. It has brought people closer by letting us share our thoughts and opinions through social networks, blogs, and forums. For example, social media has become an integral tool in decision-making for consumers looking to buy a new product. According to a survey conducted by Fleishman-Hillard [17], 50% of people who participated in the survey had looked up a product review site seeking information about brands or products. In addition to e-commerce, the Internet also provides a platform for users to share their views on any topic. There is significant interest in exploring these user-generated data, in order to analyze the general user opinion towards an entity, like a product, a brand, a political party, or a topic. This rapid growth, which is yet to be fully harnessed, has led to a new field of text mining called sentiment analysis, also referred to as opinion mining.

Sentiment analysis is the process of applying statistical, machine learning, or natural language processing methods to extract the sentiment expressed within text. Sentiment can be a user's mood in a particular situation, or an opinion or feeling towards an entity. Although research in sentiment analysis gained significance around the year 2000, coinciding with the rise of social media, earlier work on subjective analysis addresses a similar problem [18].

Standard steps involved in developing a sentiment analysis model are similar to any other text mining task. The text input is processed using methods described in Chapter 2.2.1. Text processing may also involve handling the negation of sentiment. Negation words are

called sentiment shifters since they alter the polarity of the sentiment expressed by a word. For example, “good” expresses positive sentiment while “not good” expresses negative sentiment. In advanced techniques, feature extraction is also done, which is helpful in determining the target the user’s opinion is directed towards.

One output of sentiment analysis is a binary feature, which determines the polarity of input text. A positive polarity is obtained for an input with overall positive sentiment and vice versa. A multi-point scale can also be used, similar to a star rating for movie reviews, with one end being overall positive and the other overall negative. Here, researchers have developed several lists of hand-classified words, which determine sentiment scores on a continuous scale. This is known as “a sentiment lexicon.”

2.3.1 Sentiment Analysis Levels

There are multiple ways to analyze sentiment within a document. Sentiment analysis can be classified into three types [19] based on the level at which analysis is happening.

1. *Document-level:* Document-level analysis tries to determine the overall sentiment expressed by the entire text of a document, for example, classification of movie reviews as either positive or negative [20]. This method assumes that the document discusses a single entity. It does not work well when the document refers multiple entities.
2. *Sentence level:* Every sentence in a document is analyzed to determine its sentiment. The output maybe used as input for higher levels. For example, results of sentence level analysis can be used for paragraph level, and then for document-level analysis.
3. *Entity and aspect level:* Neither document-level nor sentence level analysis try to identify the target entities a user’s opinion is directed towards. Text documents, in general, may

contain multiple entities, and each entity may have many aspects or features. For example, a review about a phone can contain more than one brand name in the document. Each phone can have many features associated with it, such as price, performance, capabilities, and so on. The idea of entity level analysis is that an opinion associated with a target is more important than an opinion that has no identified target. This provides the most detailed analysis of a text document and is also the most challenging among the three. The end result will be a summary of opinions about all entities in the document and its corresponding aspects.

2.3.2 Challenges in Sentiment Analysis

Although a reasonable accuracy in sentiment analysis has been achieved, it is still far from perfection due to the following issues [19]:

1. *Context:* A word that expresses positive or negative sentiment may have opposite meaning in other places. For example, the sentence "This movie is good," is positive due to the term "good," but in the sentence "I wish the movie were half as good as the old one," the term "good" does not necessarily express positive sentiment. In fact, the overall sentiment is negative.
2. *Ambiguity in sentiment:* A sentence may contain words that do not express any sentiment in their current usage. This situation is common in interrogative and conditional sentences. For example, the question "Is the movie any good?" does not express positive sentiment even though it has the term "good." But the question "Why did you turn down such a good offer?" expresses positive sentiment with the term "good" targeting "offer."

3. *Sarcasm:* Sarcastic statements are hard or even impossible to detect. For example, consider the following sentence “My phone worked really well! For two days!”
4. *Objective sentences:* Objective sentences do not contain any words expressing sentiment, but may imply positive or negative sentiment based on the domain under analysis. “This car uses a lot of fuel.” implies that the car has bad mileage, which is a negative statement.

2.4 Narrative Analysis and Visualization

A narrative contains an element of transformation happening through a sequence of events. It often represents a set of characters interacting either amongst each other or with the environment. Although narrative is often synonymous with “story,” narrative is more than just a set of facts. Steph Lawler [21] views narrative as social products produced by people within the context of specific social, historical, or cultural locations. Narratives are a means by which people represent themselves and their relation to the world, which includes their experience with other people or elements within the world.

Narrative analysis focuses on understanding the transformation within a story. It aims to lay out the relationship between entities, their experiences, and their relationships using features, which can be simple attributes, such as their physical location, or occupation, or advanced characteristics, such as sentiment. It is possible to employ visualization as a tool to aid the process.

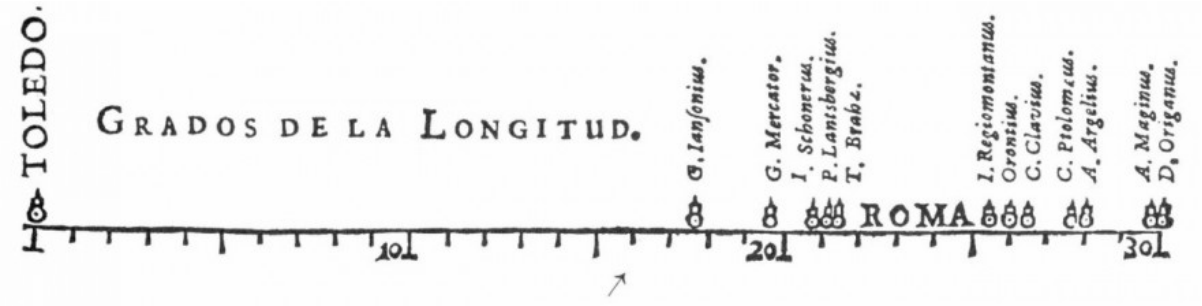

Figure 2.1 Langren's 1-dimensional graph of longitudinal distance between Rome and Toledo determined by various astronomers. The correct distance is 16º30' (value pointed by arrow) [25].

2.4.1 Visualization

The use of a one-dimensional graph by Micheal Florent Van Langren, a Flemish Astronomer, during the 17th century is believed to be one of the first visual representations of statistical data [22]. Instead of using a table, Florent used a 1-D line as a scale to show the longitudinal distance between Rome and Toledo as determined by various astronomers. The graph (Figure 2.1) shows that the estimates by the astronomers were greater than the actual value, which is 16º30', highlighted by the arrow below the horizontal axis.

2.4.2 Maps

During the early 17th century, visualization problems were mostly concerned with map-making to represent political boundaries and navigation. Over the years, maps were also used to convey data distribution across regions. In 1854, Dr. John Snow used a map of deaths due to a cholera outbreak in London to convince the town officials that the disease was spreading through usage of contaminated water and not by breathing "foul air," which was the prevalent theory during the time [22]. Tufte explains how Snow observed that cholera

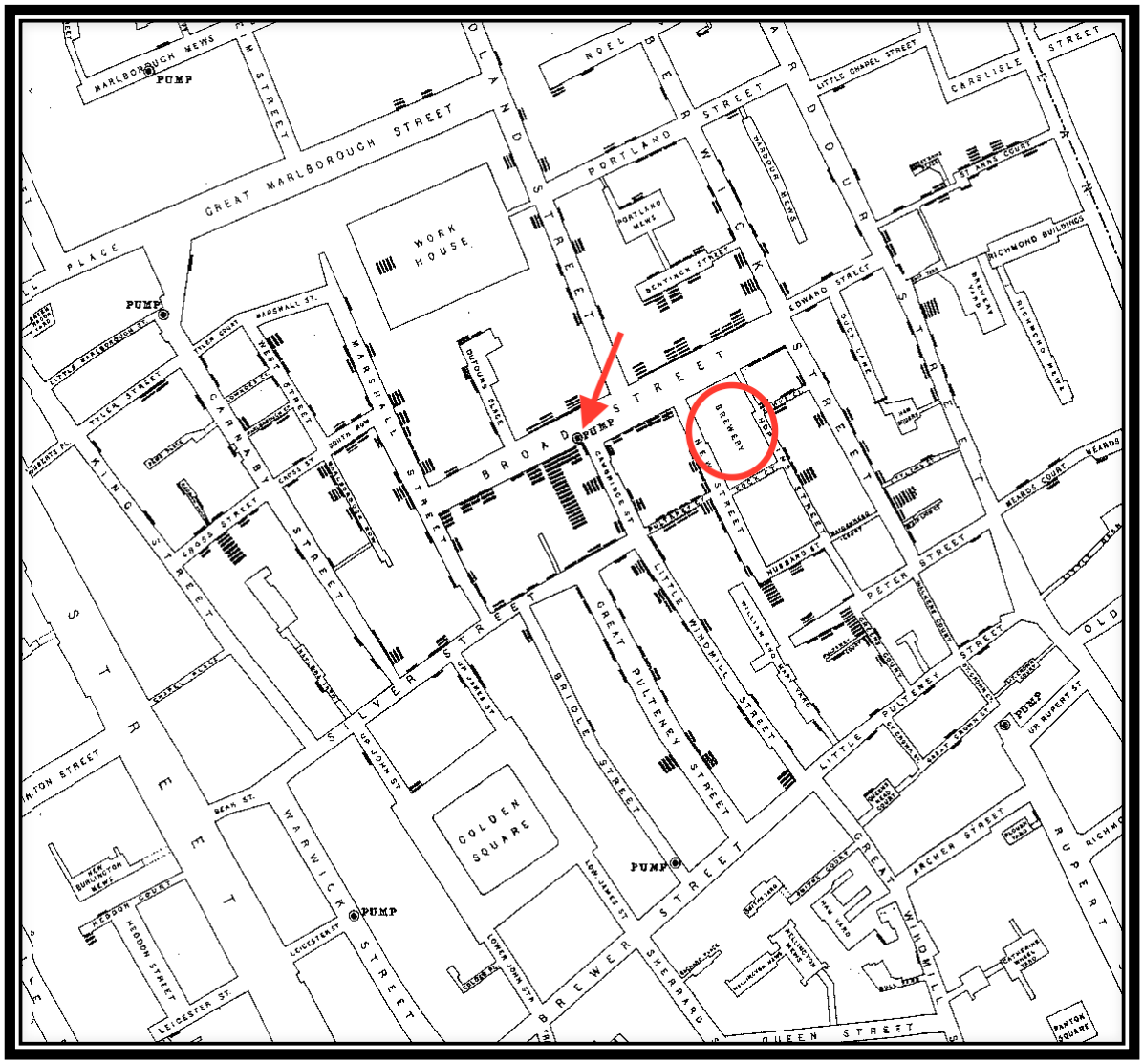

Figure 2.2 Dr. John Snow's map showing the deaths due to Cholera in London, 1854 [22].

deaths occurred almost entirely among those who lived near and drank from the Broad Street water pump.

Figure 2.2 shows John Snow's map of the cholera outbreak in Broad Street area of Central London. The locations of water pumps are represented by dots. The red arrow in the

figure highlights the Broad Street water pump. The rectangular bars represent the number of deaths due to cholera at a location. It can be observed from the map that there is a strong correlation between the number of deaths and proximity to the water pump at Broad Street. Another important observation to be made is that there were no deaths due to cholera recorded at the Brewery, which is highlighted by the red circle in the figure, although it is located very close to the water pump at Broad Street. Upon further research, Snow found out that workers at the Brewery did not use the Broad Street water pump at all, which further supports his theory.

Map based visualization is extensively used in multiple fields, such as weather, geography, census, and so on. They are also interactive in many cases, giving users a way to explore the data dynamically. Data is distinguished by means of colors or textures and sometimes both. For example, in a satellite view of the world, different types of terrain are represented using different colors. Figure 2.3 shows an interactive map visualization of the population of the U.S. during the year 2010 [23]. It shows the population percentage in each county based on the criteria of the population chosen such as race, gender, and age. A color palette is used to distinguish the ranges of population percentage. Darker colors represent higher population percentages. The number of individual ranges can be controlled. These ranges can be quantile, equal, or manually entered. There is also a classification option to choose how the ranges are formed. When the mouse is hovered over a county, a tooltip provides the percentage of the chosen population in that county. This model is helpful in providing the user with an idea about the distribution of a particular sector of people in an interactive way. Although interactive and flexible, the model represents only one variable

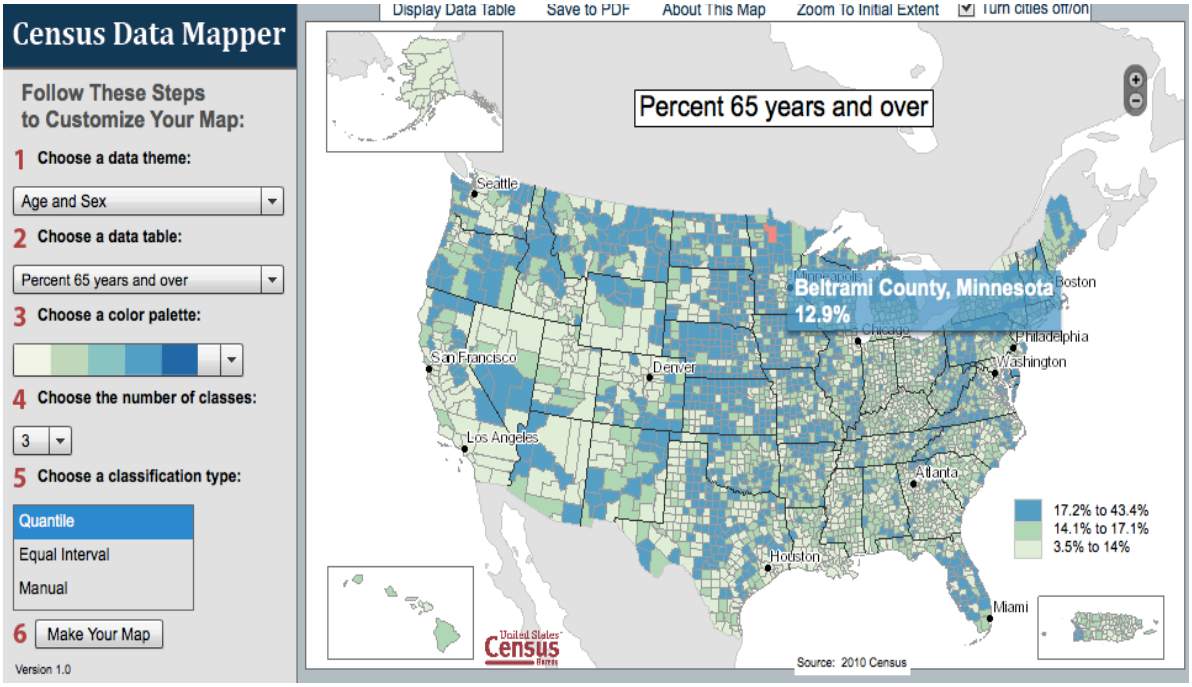

Figure 2.3 Interactive map showing census data distribution across U.S. [23].

property, which is the population percentage across the counties. If a user needs the property distribution for a different set of parameters, the model has to be re-rendered.

Another map-based visualization by Healey shows the U.S. election results for the year 2014 [24]. It presents multi-dimensional data in a single image. Each congressional district within a state is divided into four quadrants representing four elections: (1) Presidential (upper-left), (2) U.S. Senate (upper-right), (3) U.S. House (lower-right), and (4) Governor (lower-left). Color is used to represent the winning candidate's party: blue for Democrat, red for Republican, and green for Independent. Saturation is used to represent winning percentage, more saturated for a higher winning percentage. If the incumbent lost in the current election cycle, the quadrant is textured with an X pattern.

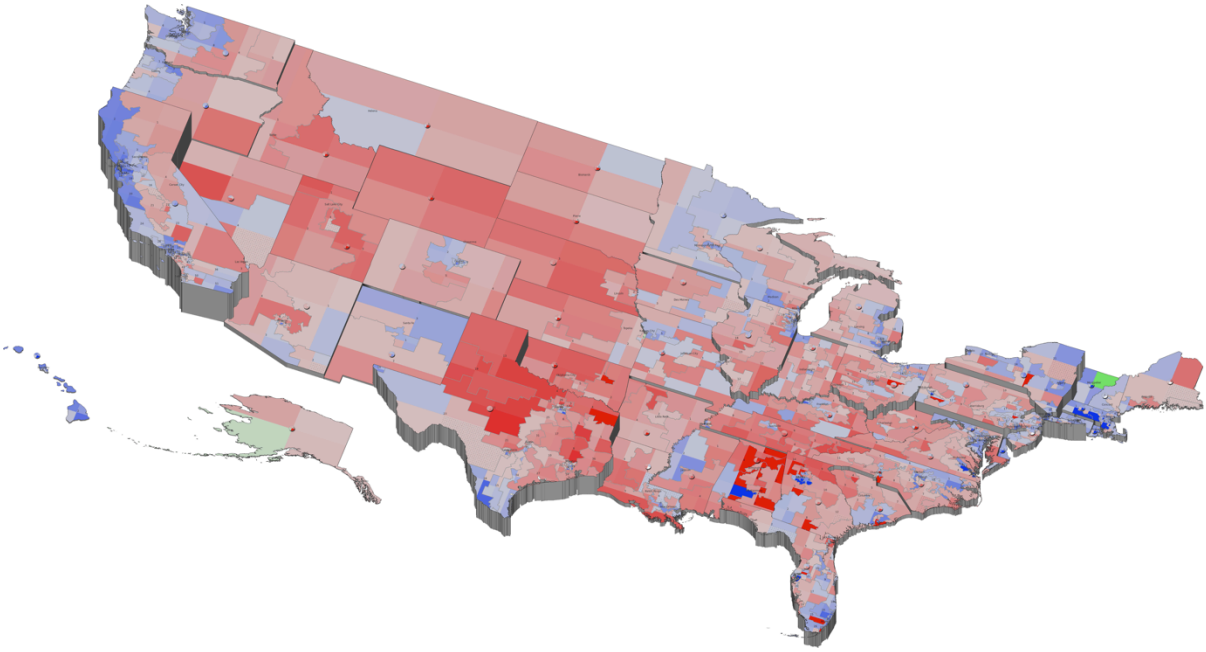

Figure 2.4 Election results visualization for all 50 states in U.S (both district and state-wide results) [24].

Every state has a small aggregate disc over it to show statewide results for the four elections. The height of each state denotes the number of electoral college votes it controls. Figure 2.4 shows a visualization for the 2014 U.S election results for all 50 states. To summarize, it shows following variables in a single image: (1) winning party, (2) winning percentage, (3) change in incumbent, (4) influence of a state in terms of its electoral votes, (5) results for each district, and (6) results for each state as a whole.

2.4.3 Charts

The 18th century saw the rise of many statistical graph types that we use today. William Playfair is widely considered to be the inventor of many common charts, such as bar charts, line charts, and pie charts [22]. This period also saw the emergence of color to add

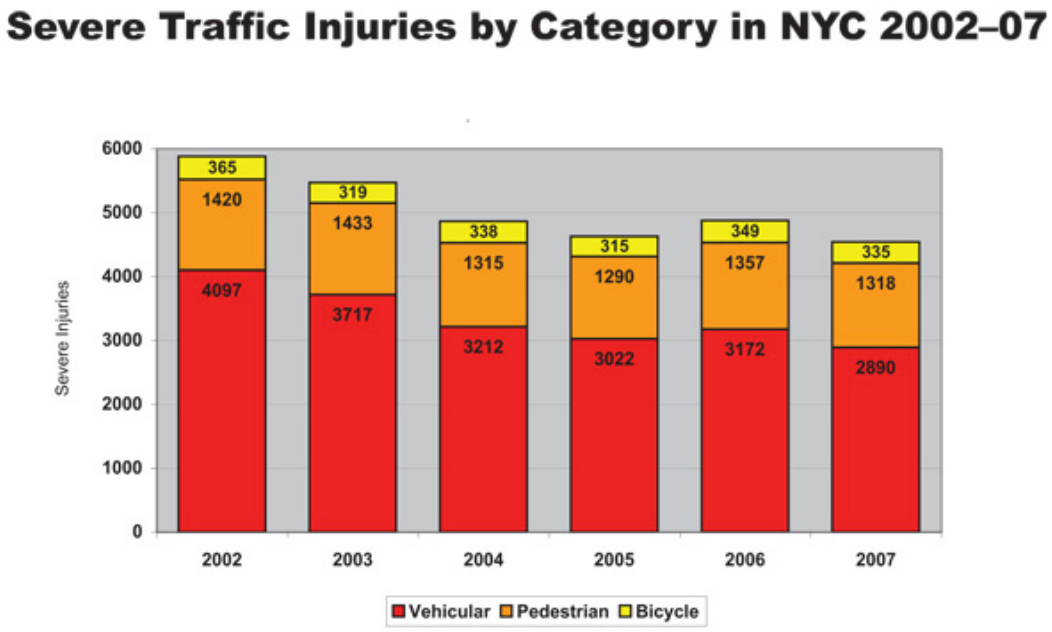

Figure 2.5 A bar chart showing traffic injuries by category between the years 2002 and 2007 [26].

additional layers of information in the charts. A bar chart uses rectangular bars, horizontal or vertical, to compare variation of a property with respect to another. One of the axes represents the value being compared, which should be ordinal data, and the other axis represents a discrete value or a set of ranges.

A bar chart can also be used to compare property variations of different elements with respect to a common axis. For example, Figure 2.5 shows a bar chart of traffic injuries from a NYC Department of Transportation report [26]. The number of injuries is shown across the vertical axis, and the year, a discrete variable, is presented along the horizontal axis. Different colors are used to represent the categories to which the injury belonged. A bar chart is one of the best ways to display univariate data, which involves only a single variable. If there are too many input data, the horizontal axis can be compressed into discrete intervals.

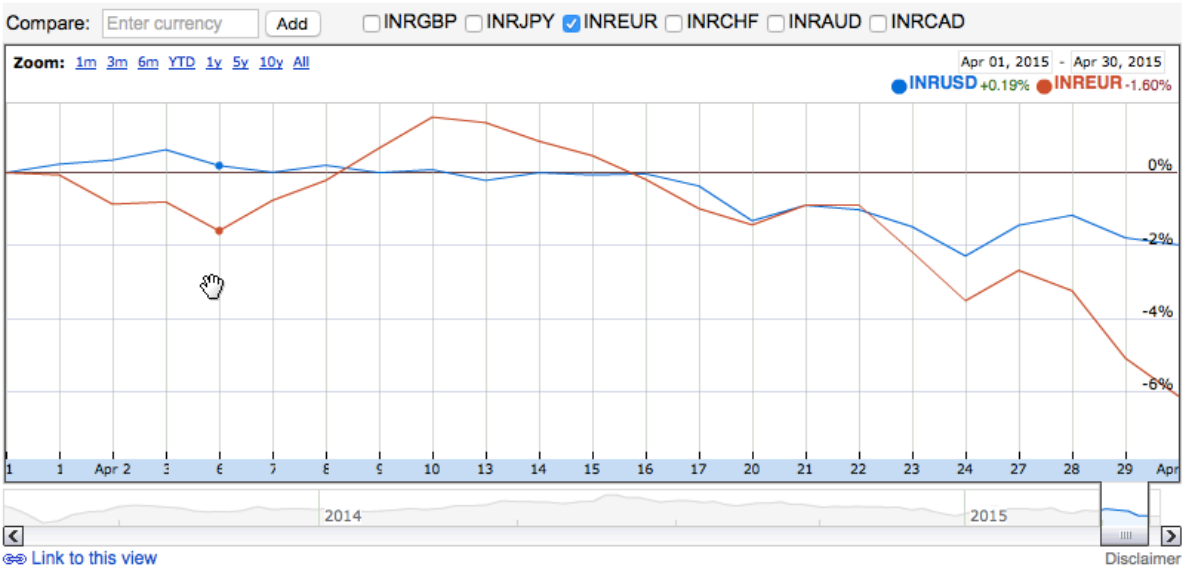

Figure 2.6 Line graph comparing the US to INR and EURO to INR conversion rate during the month of April 2015 [27].

A line graph is used to track changes in one variable versus another. It uses a vertical axis, which represents the change in one variable, and a horizontal axis, which can be time, or distance, or any ordinal variable over which the changes are tracked. Unlike bar charts, line graphs have data points that are connected together by line segments, which implies a continuous tracking variable. The data is usually ordered over the *x*-axis. For example, line graphs are commonly used to track trends in data over a time period. It can also give the user insight into the data that is not obvious from simply reading individual values, for example, small variations or trends and patterns over time. Multiple line graphs can be overlaid along a single horizontal axis to search for trends across different conditions. Figure 2.6 shows line graphs comparing the change in the currency exchange rate of U.S dollar to INR (Indian

Figure 2.7 Various correlation scenarios ranging from positive (on the left) to negative (on the right) [28].

Rupee) and Euro to INR during April 2015 [27]. The changes are recorded in percentages, starting at zero. Exact values of data points are shown when the user hovers over the line graph. It can be seen that both the exchange rates have taken a dip towards the end of the month, although USD to INR still fairs better than the EURO conversion rate.

Scatterplots are another graph visualization used to present correlation relationships between two variables. Correlation is positive when both variables increase together and negative when one of them decreases while the other increases. Figure 2.7 shows various correlation scenarios possible in a scatterplot [28]. It is similar to line graph in that it uses two axes to plot the data points, but its applications are different from a line graph. Scatterplots can be used in regression analysis to identify how a dependent variable varies with respect to an independent variable. Scatterplots can also reveal features such as clusters and outliers. Figure 2.8 shows how a scatter plot on Fisher's Iris flower dataset reveals patterns that belong to three individual species [29]. Four different features are used in the plot: (1) sepal length, (2) sepal width, (3) petal length, and (4) petal width. An array of scatter plot may be used to visualize multivariate data.

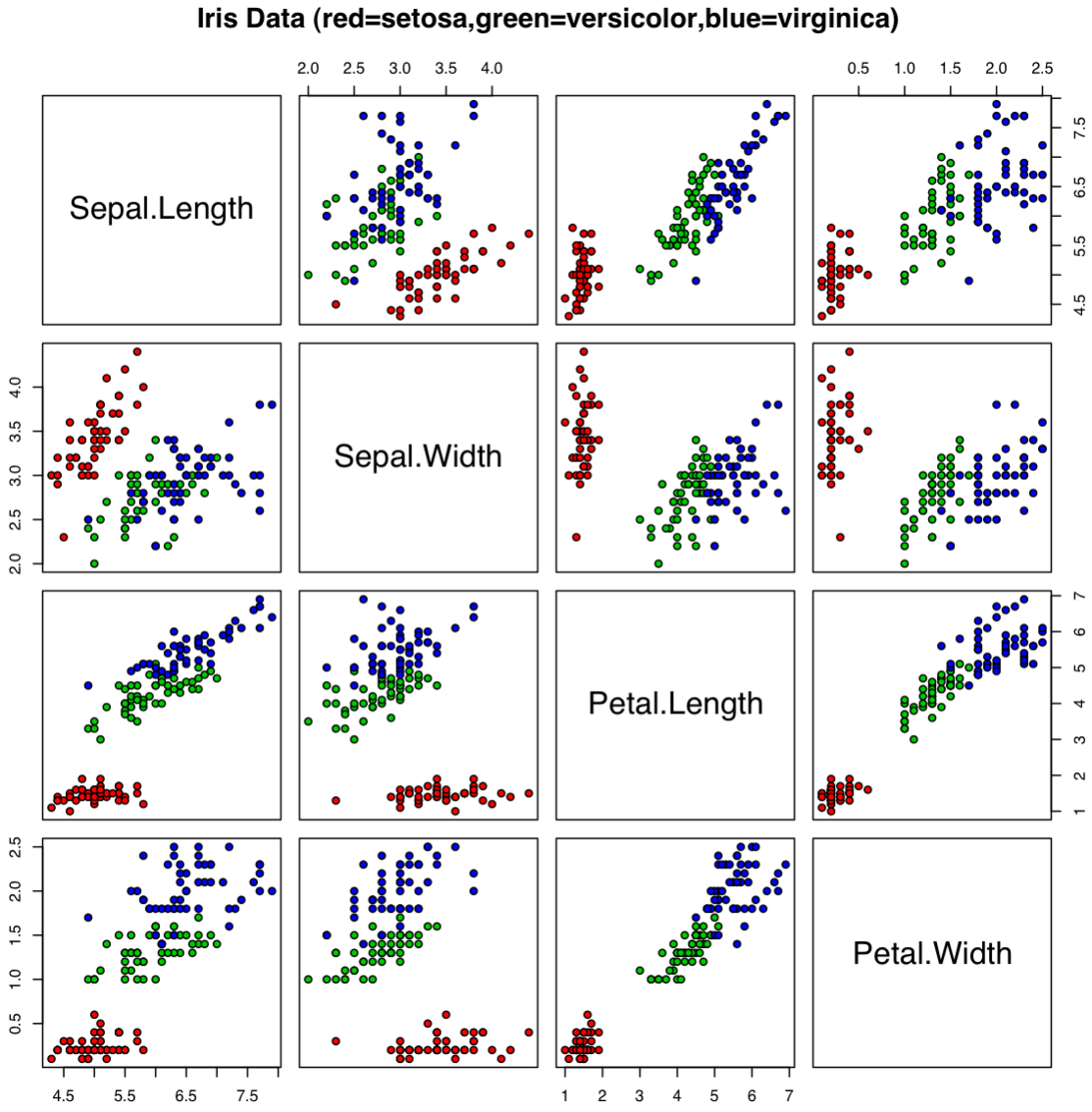

Figure 2.8 Scatter plots of Fisher's Iris flower data set [29].

2.4.4 Narrative Visualization

Understanding the flow of a narrative has been of interest for a significant period of time. Many writers have tried to identify common plot lines in stories, such as "The hero's journey" [1]. Georges Polti [30] identified and listed 36 different dramatic situations that

might occur in stories. Wojtkowski and Wojtkowski [31] analyzed the potential of storytelling in information visualization. They presented three main problems involved in designing narrative visualizations: (1) structuring the data into information, (2) identifying elements to be included in the presentation, and (3) choosing the most suitable model of visualization. They described a set of actions necessary for building a story-like visual and concluded by saying that visual storytelling might be of critical importance in providing fast and intuitive ways to explore very large data resources. Significant research has been done in the area of narrative visualization over the past two decades.

One of the earliest visualizations that adapts narrative storytelling is Charles Minard's map of losses suffered by Napoleon's army during the Russian campaign of 1812 [22]. Minard's map shows Napoleon's advance (in tan) and retreat (in black) on Moscow. It presents six different types of data (Figure 2.9). The thickness of the army's "band" represents the number of troops at any given point of time. Direction is shown by color: tan for advance, black for retreat. The path of the band roughly represents the course taken by the troops. Time and temperature during the troops' retreat is shown at the bottom of the plot. Finally, important geographic features are displayed by means of an underlying map. One can immediately realize the massive loss of life suffered by the French army and the correlation between drop in temperature, geographic features, and lives lost during the troops' retreat. Although it does not deal with text data or problems involved in structuring data into narrative, it is one of the classic examples in data visualization. Edward Tufte [22], called Minard's work "the best statistical graph ever drawn."

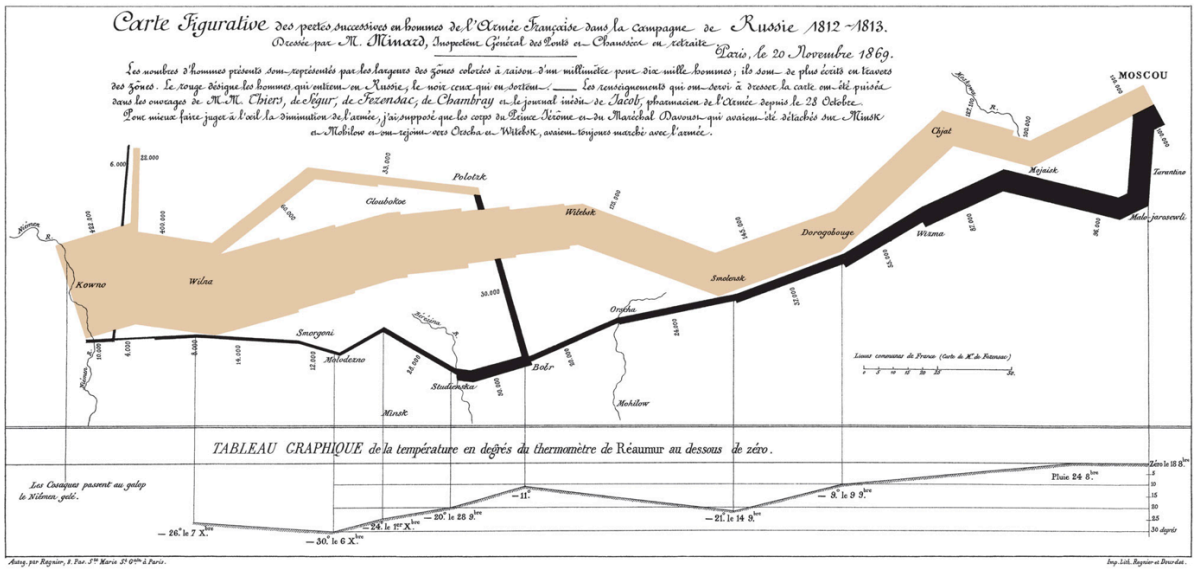

Figure 2.9 Charles Minard's map of Napoleon's Russian Campaign of 2012 [22].

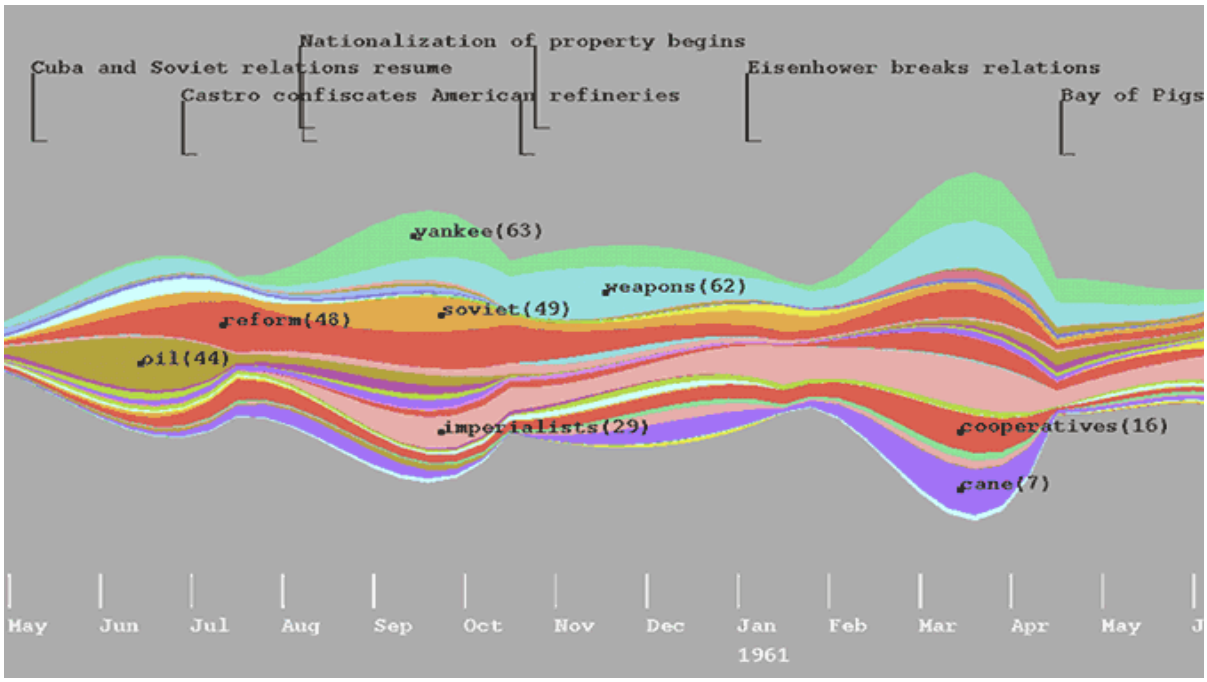

Figure 2.10 Theme River [32].

Stream graph visualizations like Theme River [32] provide contextual information by displaying the thematic strength of topics in a collection of documents over time. Documents belonging to the same theme are bundled together, with colors used to represent different themes. The vertical height of each theme changes along a temporal axis based on the number of documents within the theme at a given point of time. The result is a river like visualization comprised of multiple thematic "streams" as shown in Figure 2.10. Stream graphs are helpful in identifying thematic patterns and trends over a period of time.

Storyline visualizations are another narrative visualization technique that illustrates the dynamic relationships between entities in a story. The idea was first introduced as a hand-drawn comic titled "Movie narrative charts" in *xkcd* [2], a web comic portal. Later, Tanahashi and Ma [33] presented a set of design considerations for generating aesthetically pleasing and legible storyline visualizations. Since this technique deals directly with a narrative, their main problem involves choosing features to represent narrative relationships in the visualization. Entities are shown as individual lines and are grouped together when the entities meet at a common physical location. The timeline of the story progresses along the *x*-axis. This gives users an overview of the entire story and helps in identifying times and locations where interactions between entities contribute to critical events. When a particular line representing an entity ceases to exist, that may mean that entity's location is undefined, or the character is removed from the story. Figure 2.11 shows a storyline visualization of Lord of the Rings. Yellow line represents the path taken by the ring. Entities are color coded to represent their kind, for example, grey lines are used for wizards, blue lines are used for elves, and so on. When entities of the same kind co-occur at the same locations for most of

the story, they are bundled together. For example, the thick black layer at the bottom represents Sauron's army. Significant events in the storyline, such as a battle, are emphasized by using a colored background.

Tweet Viz is a twitter visualization tool developed by Healey and Ramaswamy [34]. Tweets are collected based on an input query term and visualized in numerous ways, such as by sentiment, by frequency of words, by topics, and so on. It implements dictionary-based sentiment analysis on Twitter data using a sentiment dictionary. Pleasant tweets are shown in green and unpleasant tweets in blue. Each tweet is represented as a circular element, with its size and opacity proportional to the confidence of the sentiment estimate. The sentiment results are presented as various two-dimensional charts and maps, for example, in a sentiment scatterplot, with horizontal axis representing the overall pleasure of the tweet (right for pleasant and left for unpleasant) and the vertical axis representing the emotional state (top for active and bottom for subdued). Figure 2.12 shows the sentiment results for the query term "batman." By clicking on a tweet, the user can read its content from and identify keywords that were used to estimate its sentiment. The visualization includes other representations, including a tag cloud for each sentiment quadrant (happy, relaxed, sad, upset), and a timeline view bar chart that shows tweet counts by sentiment over time.

Bilenko and Miyakawa [35] created a story visualization that shows both a character interaction graph and a sentiment bar plot in chapter order. The character graph presents all participating characters in a radial layout. A link between two characters represents their appearance in the same chapter within a specified number of sentences. The thickness of the link is proportional to the number of such co-occurrences and determines the strength of

relationship between characters. The characters of similar type are grouped together, for example, in *The Hobbit* visualization shown in Figure 2.13, character types include hobbits, elves, dwarfs, and so on. The arc path of the link is shorter and flatter if the characters are from the same group. Each bar in the sentiment plot represents a sentence in the novel arranged in chronological order. The direction of the bar defines its polarity and its height defines the level of sentiment, with zero as the center of the graph. Although character information is useful, sentiment information only provides an overview of sentiment across the entire story. No option is provided to track sentiment at the chapter level, which could provide more information about outcomes based on entity interaction.

2.4.5 Information Visualization

Information visualization focuses on methods to visualize more abstract data. This requires defining layout for the data prior to visualizing it. For example, a treemap is an information visualization technique used to represent hierarchically structured data by means of nested rectangles. It was developed by Ben Shneiderman in 1990 to visualize hard drive directory structures [36]. Each branch of the underlying tree hierarchy is assigned a rectangle. Sub-branches produce inset rectangles inside a parent rectangle. The area of each rectangle is proportional to a data value, for example, the total size of the files in the given directory or sub-directory. Color schemes can be used to distinguish between branches.

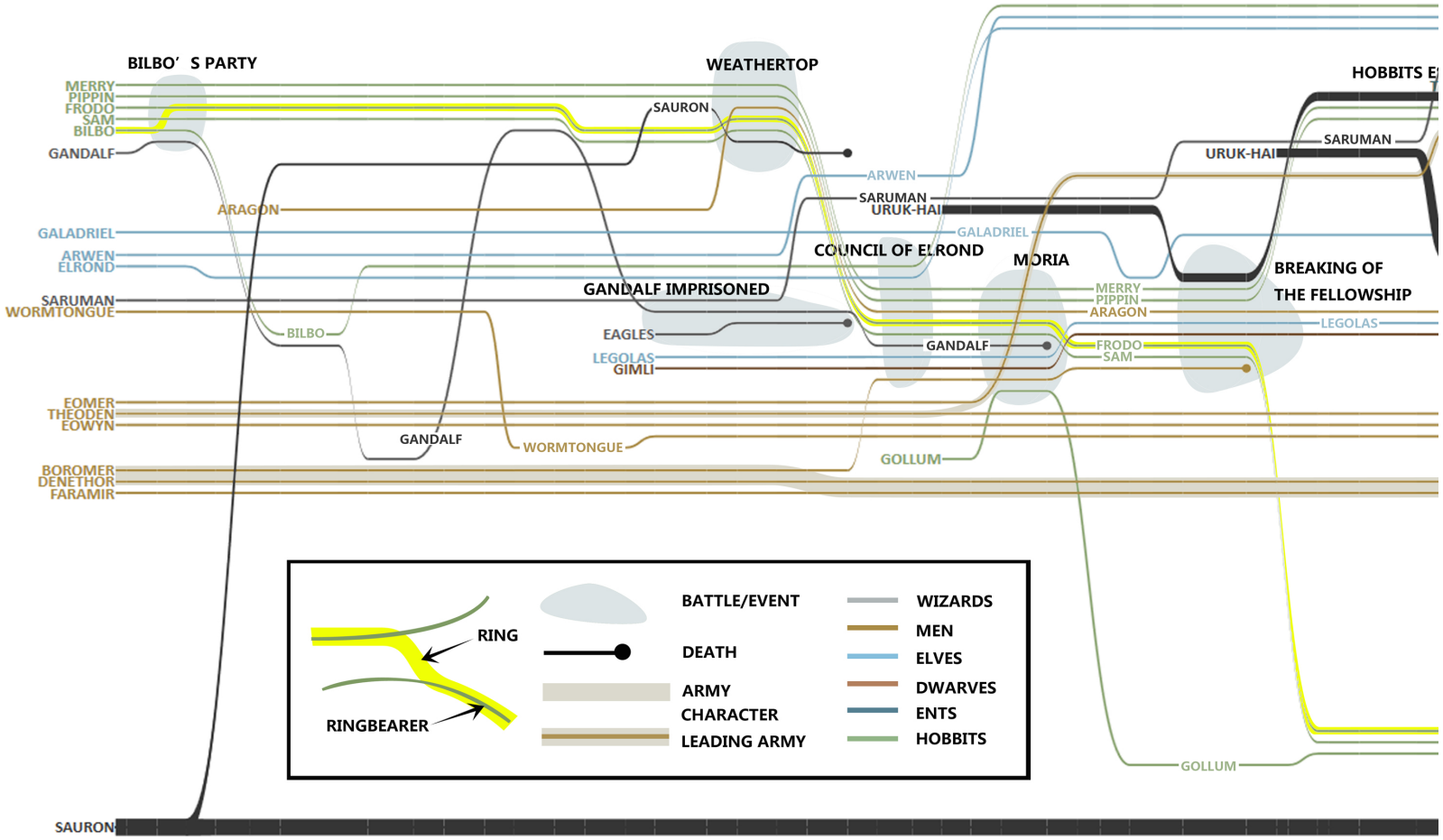

Figure 2.11 Storyline visualization of Lord of the Rings (incomplete since cropped to fit in a single page) [33].

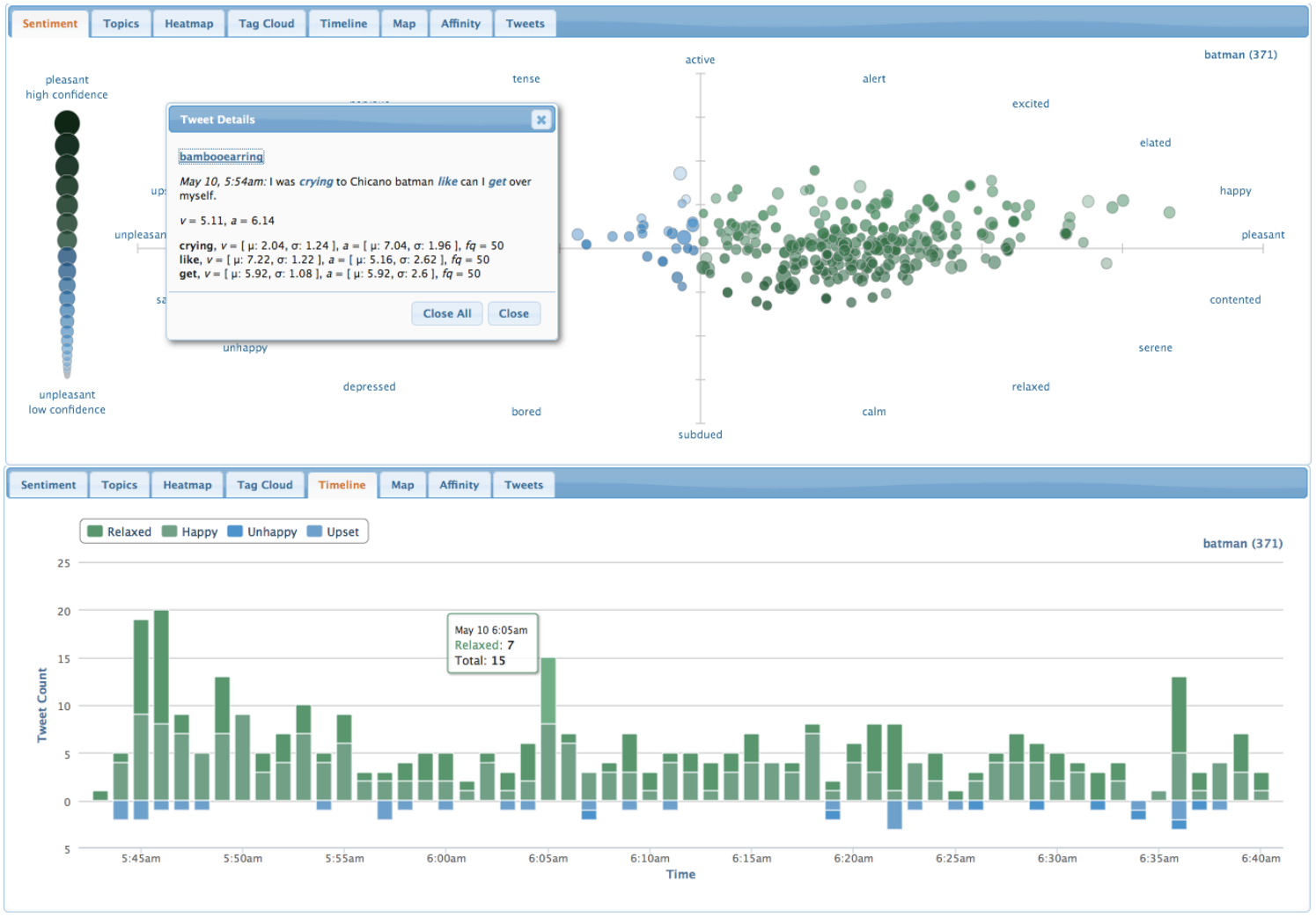

Figure 2.12 The figure on top shows the sentiment visualization tab of Tweet Viz based on the search query “batman.”

One on the bottom shows the timeline view of these tweets [34].

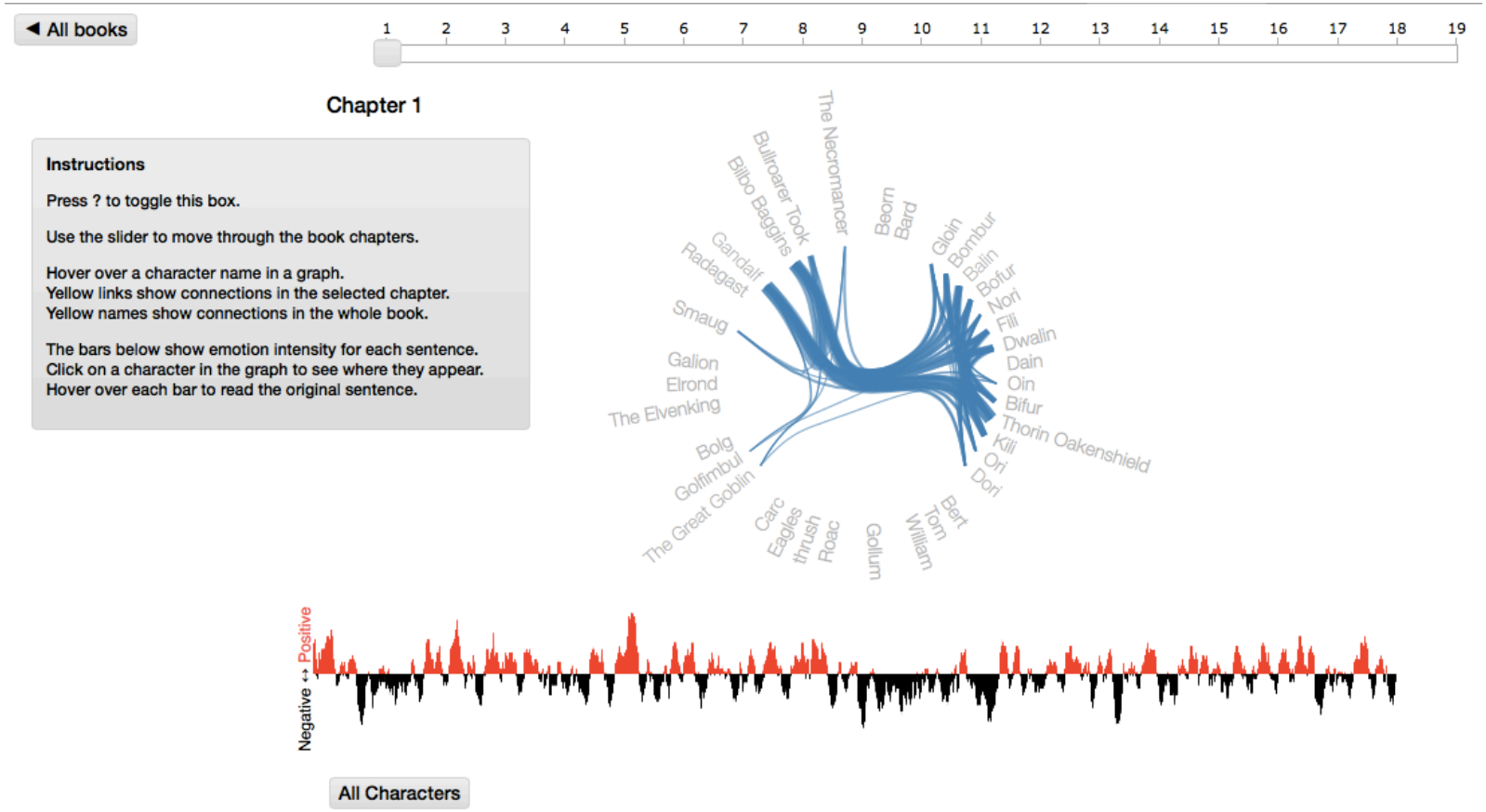

Figure 2.13 Visualization of Narrative Structure by Bilenko and Miayakawa [35].

Figure 2.14 shows an example of a treemap from an article in the *New York Times* that visualizes the market capitalization of 29 financial firms on Oct 9, 2007 [37]. The size of the rectangles represents the market capitalization value of the companies. Colors are used to distinguish between the types of companies, for example, blue for National Commercial Banks, and grey for Asset Managers and Investors, and so on. This particular treemap has no hierarchical structure, as there are no sub-branches within. Treemap can be useful for comparing different branches within the tree structure to identify patterns or anomalies.

Information visualization can also be applied to text documents. Identifying the most significant words from a text corpus gives useful information about the data. Word clouds (tag clouds) are a visual technique that represents words from a corpus based on their importance. The importance can be any statistical measure such as frequency or term weight. The size and color of the words can also be varied based on their different properties. Wordle [38] is an online tool developed by Jonathan Feinberg to generate word clouds from text input. Figure 2.15 shows the word cloud generated by providing the text from Wikipedia home page [39]. The size of the word's font represents how frequently it occurred in the text corpus. In this particular visualization color and rotation of the words are randomized, but these properties can also be modified based on certain criteria, such as part-of-speech, to convey more information about the words. The words do not follow a strict ordering and are packed tightly in order to be space efficient.

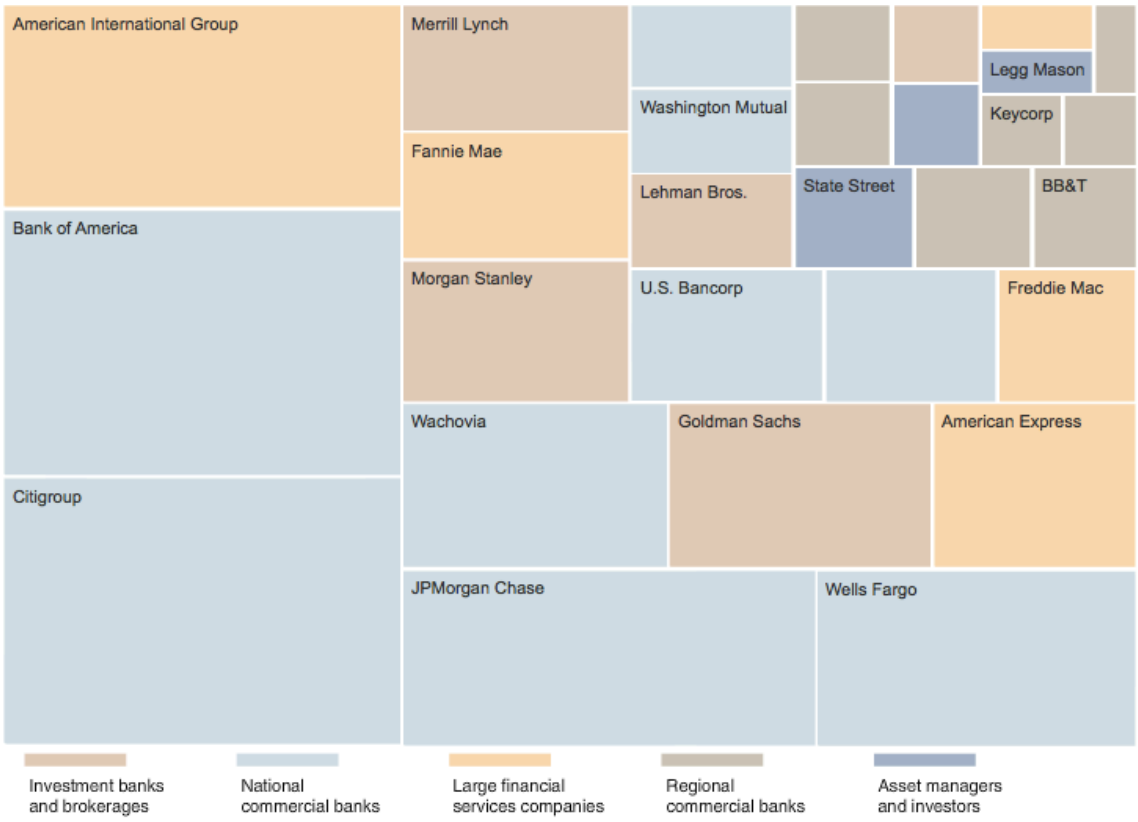

Figure 2.14 An online New York Times article using Treemap to show the market status of a selected 29 financial firms [37].

Figure 2.15 Word cloud of text from Wikipedia Home Page [38] [39].

Chapter 3

Sentiment Estimation

One important task in existing sentiment analysis algorithms is the creation of training sets, which are used to build statistical or machine learning models. Practitioners can spend significant time hand-classifying text data to create training sets. The algorithm presented in this thesis proposes a method to automatically create training sets using a bag-of-words sentiment analysis approach, which compares words within a given text against a sentiment dictionary. The dictionary, also called "a sentiment lexicon," contains common words that express sentiment, with one or more sentiment scores for each word. Several sentiment dictionaries have been created. Some of the popular dictionaries are:

1. SentiWordNet, an open-source lexical resource for opinion mining [40]. It is based on WordNet, a large lexical database of English language developed by Fellbaum [41]. Words that have similar meaning are grouped together to form "synsets" in WordNet. SentiWordNet associates each synset with numerical scores describing how objective, positive, and negative its words are.
2. Affective Norms for English Words (ANEW) contains a set of emotional ratings for a large number of English words, developed by the Center for Emotion and Attention (CESA) at the University of Florida [42]. Words are scored by pleasure, arousal, and dominance. The original version of ANEW contained 1,034 words. An extended version was recently built by Warriner *et al* [43]. It contains 13,915 words.

3. Sentistrength [44] is a lexicon-based classifier that estimates positive and negative sentiment for short pieces of text. Sentistrength was specifically designed to analyze social media text, so it also works on informal language and social media properties, such as emoticons. Sentistrength tries to identify the sentiment expressed in an input text by providing estimates for both positive and negative sentiment, rather than calculating an overall polarity. It can report output in three different formats: (1) binary (positive or negative), (2) trinary (positive or negative or neutral), and (3) single scale (ranging from -4 for most negative to +4 for most positive). In addition to a list of human-annotated words for sentiment analysis, Sentistrength provides additional features such as a spelling correction, a negation word list, sentiment polarities for emoticons, and so on.
4. Profile of Mood States (POMS) is a psychological test developed by McNair [45] to assess the mood of an individual. It identifies six different states of mind: (1) tension-anxiety, (2) depression-dejection, (3) anger-hostility, (4) vigor-activity, (5) fatigue-inertia, and (6) confusion-bewilderment. POMS uses a total of 65 adjectives rated on a five-point scale. Individuals taking the test provide a rating for each of these words, which are then used to calculate a Total Mood Disturbance and other metrics. Bollen *et al* [46] used an extended version of the original POMS word list to perform sentiment analysis on tweets. They used the POMS scoring method to match words extracted from each tweet to the set of adjectives for each of POMS' mood dimensions, producing a sentiment estimate for each dimensions.
5. Bollen *et al* [47] later developed a second extended version of POMS, called POMS-ex that is more suitable for a large text corpus. POMS-ex extends the original set of 65

adjectives in POMS to 793 words, made up of synonyms and semantically related words of the original set.

6. Hu and Liu [48] have developed a list of 6,800 words categorized as positive or negative opinion words, which they first presented in their paper on identifying opinion sentences in customer reviews and classifying each of them as positive or negative.

We chose the ANEW dictionary because it is a popular resource and has proven to perform well in sentiment analysis on short pieces of text, such as tweets [34] and micro blogs [49]. The sentiment scores for the ANEW words are provided on a continuous scale from one to nine. Both the average rating and the standard deviation of ratings by all volunteers who evaluated each word are given for the pleasure, arousal, and dominance dimensions. This chapter provides a detailed overview of dictionary-based sentiment analysis and describes how it can be used to construct a training set as input for more advanced sentiment analysis algorithms. Finally, we discuss how the sentiment of a single document is estimated based on these results.

3.1 Automated Training Set Generation

The main goal of training set generation is to construct a training set automatically that contains estimated sentiment results for each text entity. If needed, this training set can be efficiently refined. The refinement process, though manual, is much less time-consuming than generating an entire hand-classified training set. The cause of approximations in the automatically generated training set is due to the assumption that the presence of any word from the sentiment lexicon alone is sufficient to determine sentiment. Issues that were

previously described in Chapter 2.3.2, such as word ambiguity, sarcasm, and so on, as well as domain specific terminology, are not considered.

A mathematical model computes a score for the input text based on the keywords present in the ANEW dictionary, which provides sentiment scores for each word for three emotional dimensions: valence, arousal, and dominance. The ANEW words were selected based on previous research that identified them as good candidates to express emotion. We considered only the valence score of a word while calculating the sentiment since it corresponds to a pleasure rating, which directly relates to the polarity of sentiment expressed by the word. The SAS® Sentiment Analysis tool we are using is currently only capable of considering polarity in its training. Each word in the ANEW dictionary is rated on a scale from one to nine. To construct the dictionary, volunteers were asked to read a text corpus and provide a rating for each occurrence of an ANEW-recognized word. The final rating of a word is the mean and standard deviation of all the ratings by all volunteers. An example of an ANEW word's rating is:

education :

$$v = (\alpha : 6.69, \sigma : 1.77)$$
$$a = (\alpha : 5.74, \sigma : 2.46)$$
$$d = (\alpha : 6.15, \sigma : 2.35)$$
$$f = 214$$

where v is valence, a is arousal, d is dominance, and f is the number of ratings of the word. α is the average and σ is the standard deviation of the given score. A lower standard deviation indicates more consensus in the score.

The baseline algorithm in a dictionary-based sentiment analysis involves the following two steps: (1) text pre-processing and (2) sentiment scoring.

3.1.1 Text Pre-Processing

The input text data is first split into smaller pieces of text. This is done to ensure that each block contains a single sentiment. For example, if a document is a chat transcript, it could be split into statements made by each entity. If it is a blog or news article, it could be split into sentences or paragraphs. This is based on the intuition that information contained within a single text block is coherent and related. It also ensures better sentiment results since dictionary-based sentiment analysis is designed for shorter piece of text. The overall sentiment of a document is calculated based on the sentiment scores of individual blocks or segments that make up the document. This approach provides more accurate sentiment results than analyzing the entire text of the document. Before keyword search is performed, stop words and punctuations are removed, and the text is tokenized to produce a word vector present in the input.

3.1.2 Sentiment Scoring

Each word vector is searched for the presence of ANEW-recognized words. We assume that the minimum number of ANEW-recognized words required for an input text to express sentiment must be two or more. If it is less than two, then the input is ignored for analysis and considered neutral. The sentiment score of valid input is calculated as the weighted average of the sentiment scores of all the ANEW-recognized words in the input word vector.

$$\text{Sentiment score} = \frac{\sum v_i/\sigma_i}{\sum 1/\sigma_i} \quad (1)$$

where v_i is the valence of the i^{th} word in the list of ANEW-recognized words, and σ_i is its standard deviation.

Table 3.1 Valence scores for ANEW-recognized words in the text.

Words	Valence	
	Average	Standard deviation
sea	4.95	2.79
food	7.65	1.37
see	6.1	2.19
eat	7.47	1.73

If the resulting score is greater than or equal to 6.0, it is classified as positive. Input with a sentiment score of 4.0 or less is classified as negative. Scores between 4.0 and 6.0 are classified as neutral.

For example, in the following text, "I am on a *sea food* diet. I see *food*, and I *eat* it," the three ANEW-recognized words are *italicized.* Table 3.1 shows the ANEW valence scores of the words. Using Formula 1, we obtain a sentiment score of 7.02, and, the text is classified as positive.

The standard baseline algorithm for sentiment analysis using a sentiment dictionary requires a statistical or machine learning model to obtain better output performance. But, since we only require the results to bootstrap the process of sentiment analysis using a follow-on tool, we can terminate the algorithm at this point. Another important factor is that negation of sentiment is not handled in dictionary-based sentiment analysis. Hence, one has to make sure the follow-on tool has the ability recognize negation.

As a result of training set generation, we obtain a list of text blocks, which may be sentences or paragraphs, and their sentiment classification, which can be positive, negative, or neutral.

3.2 SAS® Sentiment Analysis

Although dictionary-based sentiment analysis provides reasonable results, to test for improved performance we need tools that have more advanced features, such as probabilistic Bayesian models, rule-based classifiers, and so on. In this implementation we have chosen the SAS® Sentiment Analysis (SA) tool. It is a state-of-the-art system and one of the best performing tools currently available. SA has an option of using a statistical modeling technique to identify significant words from the text input based on their frequency of occurrence. We take advantage of this feature to extract domain specific keywords from the text input. These words can further be categorized into a positive or negative list or removed from consideration. SA has a rule-based classifier option, which can be used to detect phrases in the input text matching a certain pattern. Pattern matching considers all the words from the positive or negative lists, or a combination of both. It can also detect parts-of-speech like adjectives, adverbs, nouns and so on. Multiple rules can be written to detect phrases within the input text, and each rule can be assigned different weights to represent its significance.

3.2.1 Building a SAS® Sentiment Analysis Model

Sentiment analysis using the SAS® SA tool is a two-step process.

1. *Statistical modeling.* SA includes a statistical component, which is used for discovery of significant words to facilitate rule writing. The training set from the ANEW analysis is used to build a simple statistical model that classifies the input text as positive or negative based on a naïve Bayes method. SA automatically attempts different text normalization models and feature ranking algorithms to identify the best possible Bayes model.

2. *Rule-based classification.* The sentiment keywords identified by the statistical model for classification are imported and used for building a rule-based classifier model. The number of words to be imported can be controlled. We use approximately 250-400 words. Words that are insignificant and do not convey any sentiment information are removed, for example, numbers, names, addresses, and so on. The words are then split into two lists: "PositivePhrases" and "NegativePhrases," based on their sentiment. A separate list called "Negation" is created that contains the words that negate the sentiment of a word, such as "not," "don't," and so on. Reckman *et al* demonstrate the application of rule-based classifier for sentiment analysis and also give some example rules [51]. They describe rules as patterns that match words or sequence of words. Matching happens left-to-right and longer matches take precedence over shorter ones. The advantage of rule writing is that errors in the output can be targeted directly, making it easier to refine the model. This process can be performed iteratively until the rule-based classifier model provides the best possible results on a test data set.

SA comes with a built-in part-of-speech tagger. Rules can require the words in a pattern to match a particular part-of-speech. An "**@**" sign can be used to enable morphological expansion. This denotes that the rule matches any form of the word. For example, "win@" matches win, wins, won, and winning. Rules can be organized into lists so that one rule refers to another.

We will discuss some basic rules that were used to detect the sentiment of different phrases. To refer a rule from a list "listname," a `_def{listname}` keyword is used. For example, to identify the presence of positive or negative words, we used the rule

`_def{PositivePhrases}` and `_def{NegativePhrases}` respectively. The following rule is used to identify negation of sentiment,

```
_def{Negation}_def{PositivePhrases}
```

This rule matches phrases that contain any words from the "Negation" list followed by any words from "PositivePhrases." The negation rule for negative sentiment is similar, with "NegativePhrases" used in place of "PositivePhrases". To detect co-occurrences of words in a single sentence a `SENT` operator is used.

```
(SENT, (DIST_4, "_a{_def(Price)}", "_a{_def{Reduce}}"))
```

This rule detects if the prices, defined by the rule list "Price," were reduced, defined by the rule list "Reduce." The pattern matches only if the words occur within a distance of four words from one another within a common sentence. This could be identified as positive sentiment, for example, when we are dealing with a domain involving consumer products, since this would make a buyer happy. Another operator similar to `SENT` is `ORDDIST_n`, which matches a pattern only if words from the lists occur in a particular order. *n* corresponds to the distance within which words can co-occur.

It is possible that some patterns maybe detected by more than one rule. SA gives users an option to assign weights to the rules to overcome this problem. The output can have three different values: 1 for positive sentiment, -1 for negative sentiment, 0 when positive and negative weights cancel each other, and "NA" if none of the patterns are detected in the input. We classify an output of 0 or "NA" as neutral. SA also provides a confidence rating for its output, which determines how confident the tool is about a specific result.

3.2.2 Cross-Validation

When a model is trained and tested on the entire training set, it is only trained to repeat the labels of samples it has seen. It will perform poorly when predicting unseen data. This situation is called overfitting. In order to avoid this, when training a supervised machine-learning algorithm, it is necessary to hold out part of the set used to train the model, then use this holdout subset for testing. This is achieved by a resampling technique called cross-validation. The sample set used for training is separated into *n* subsets. The hold out process is then repeated *n* times, each time choosing one subset as the test dataset and the remaining data for training. In this way every sample is included in the test dataset at least once and in the training set *n*-1 times.

Figure 3.1 shows a pictorial representation of *k*-fold cross-validation, with *k*=4 [50]. We used *k*-fold cross-validation with *k*=10, which is a commonly used approach for training supervised machine learning algorithm. The results from ANEW-based sentiment analysis were split into ten different subsets. 10-fold cross-validation was then performed to build ten different SAS® sentiment Analysis models. Each time, the test dataset was checked for misclassification by ANEW-based analysis and corrected. This allowed the SAS® model to be updated accordingly. Most errors were due to the issues mentioned in Chapter 2.3.3, such as different contexts for a word in current domain, negation of sentiment, and so on. Also, Sentiment Analysis tool extracts keywords from the training set based on a statistical model. Words that occurred with high frequency but conveyed no sentiment, such as numbers and nouns, were deemed insignificant. Finally, all the models were merged to provide a single model, which was again verified with all the test datasets and improved further if necessary.

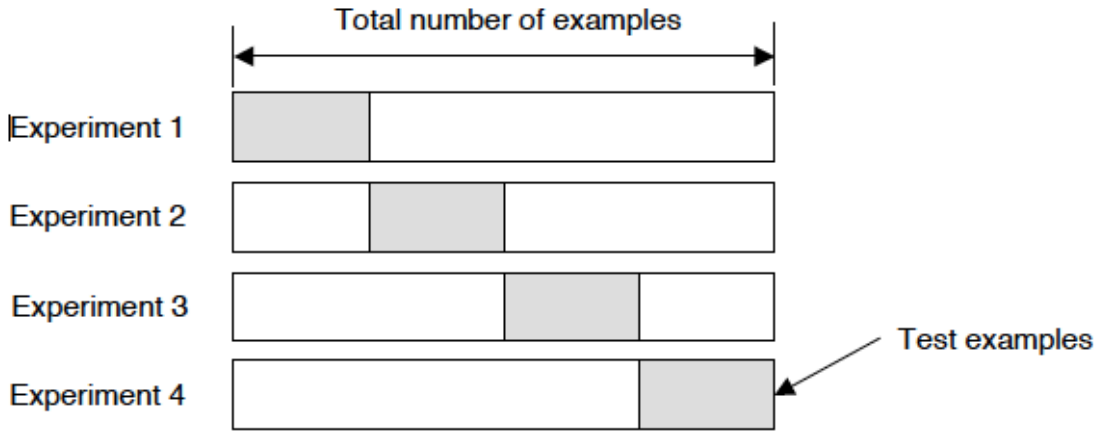

Figure 3.1 Pictorial representation of *k*-fold cross-validation technique, with $k = 4$ [50].

3.3 Estimating Overall Sentiment of a Document

Many existing algorithms estimate a document's sentiment based on the entire text of the document. This usually results in an inaccurate or meaningless estimate, since they assume there is only one entity in a document, which is not usually the case for many unstructured text data, such as blogs, chats, discussions, and so on. These kinds of data may have different users expressing opinions on different entities. For a better understanding of a document's sentiment, the sentiment variations within it should be considered. This was the reason our documents were divided into shorter text blocks or events. A document's sentiment is then represented in terms of its events' sentiment. The overall sentiment of the document is calculated by identifying the most dominant sentiment present within the document. This measure of overall document sentiment together with the sentiment of the events within the document provides a better understanding of document-level sentiment.

Table 3.2 Sentiment results of sentences in a sample conversation.

Sentence Id	Sentence	Sentiment	Confidence Rating
Person 1	Hey, did you checkout the new Apple Watch? It is so awesome.	Positive	0.2
Person 2	Yeah! I did, but I didn't like it.	Negative	0.2
Person 1	Why not?	Neutral	0.0
Person 2	Well, it's expensive and not so great.	Negative	0.2951
Person 1	Yeah. But, I have the money, and I like the way it looks.	Positive	0.3846
Person 2	Oh, good for you then. Have fun.	Positive	0.3846

Table 3.3 Average confidence ratings of each sentiment in the sample conversation.

Sentiment	Average Confidence Rating
Positive	0.3230
Negative	0.2475

Table 3.2 shows the sentiment results of sentences in a chat conversation between two people based on our automated training of SA. Since neutral sentences do not convey any sentiment information, a confidence rating of 0 is assigned. From the average confidence ratings of all positive and negative sentiments within the document (Table 3.3), it is obvious that a positive sentiment is dominant. Hence, the document is assigned a positive polarity.

Chapter 4

Narrative Thread and Visualization

In this chapter we discuss how a narrative thread is built from a document using the results from sentiment analysis. A line graph is used to visualize the narrative within a document. Line graphs were chosen because they are an effective way to visualize small changes across a timeline and aid in the identification of critical events that produce sentiment transitions. Line graphs are also a well-known visualization framework, which significantly increases their accessibility to user without formal visualization training. Finally, line graphs can be used in both dynamic, interactive environments like computer programs and static environment like printer reports. Sentiment keywords present in each event are used to tag the block element, providing insights to the user to determine the cause for changes in sentiment.

4.1 Structuring a Narrative Thread

Interactions between entities result in outcomes due to changes in one or more features associated with a narrative's events. We have analyzed one such feature here, the sentiment of events. Sentiment conveys significant information regarding the nature of an interaction. For example, a conversation between two persons might have started off friendly, but ended in a hostile manner. Another example is a product review with mixed opinions. Sentiment transitions answer important questions regarding the nature of an interaction, such as why there was a difference in opinion, and what or who caused it.

A narrative structure demands a sequence of events and its properties. We discussed how a document is divided into events in Chapter 3. We use the events within a document and their sentiment to build a narrative. An obvious way to visualize a narrative is to plot the sentiment results of events within a document in the order of their occurrence. To do this, we implement a line graph to visualize a narrative thread as it unfolds over time.

4.2 Constructing a Line Graph

Given events and their corresponding sentiment and confidence for all the documents in a document collection, we construct a line graph with sentiment scores plotted on the *y*-axis, and the events within a document was ordered along the *x*-axis.

Positive and negative sentiments have to be emphasized since they are significant. Using fully saturated colors alongside a less saturated color can achieve this effect. Grey was used to represent neutral sentiment. Blue was used for positive sentiment and red for negative sentiment. Both blue and red are highly saturated and provide sufficient contrast to represent opposite sentiment polarities. Using blue and red also avoids perceptual issues with viewers who are red-green color blind, a common issue affecting up to 10% of the general population. Horizontal dotted lines at 6.0 and 4.0 respectively, are used to represent the thresholds for positive and negative sentiment. These threshold values are same as the thresholds used during ANEW dictionary-based sentiment analysis for generating training sets. Since the results from SA are in categorical form, values for positive and negative sentiment are set to sentiment scores of 7.5 and 2.5, respectively. This was done to maintain consistency in scoring between ANEW and SA. Thresholds were displayed using their

corresponding colors (blue for positive, red for negative) to clearly demarcate the regions for different sentiments.

Events are represented as circles, with colors corresponding to their sentiment. Neutral events were reduced in size to reduce their visual significance. The events in a narrative thread are joined together by a yellow line that tracks the flow of sentiment through the document. Figure 4.1 shows the line graph for a narrative built from the sample chat in Table 3.2. It can be observed from the graph that the conversation starts positively, but immediately drops below the negative threshold. It enters the neutral zone once, then drops below the negative threshold again. Finally, it rises above the positive threshold in the next event and stays there until the end of conversation. Three important observations can be made from the narrative thread visualization:

1. The conversation started on a positive note.
2. There were two sentiment transitions: positive to negative sentiment and negative to positive sentiment within the narrative.
3. The conversation ended positively.

We identified the two sentiment transitions within the narrative and analyzed the events involved for words that express sentiment using the ANEW dictionary. Among the ANEW-recognized words, one or two words with a maximum sentiment rating were chosen and tagged beside the events. In the current example, for the first sentence, the term “awesome” is tagged as shown in Figure 4.1, and for the next event, “like” is tagged. These keywords give the user an idea that something recognized as awesome by the first person wasn’t favored by the second person. Although the word “like” gives a positive sense, its

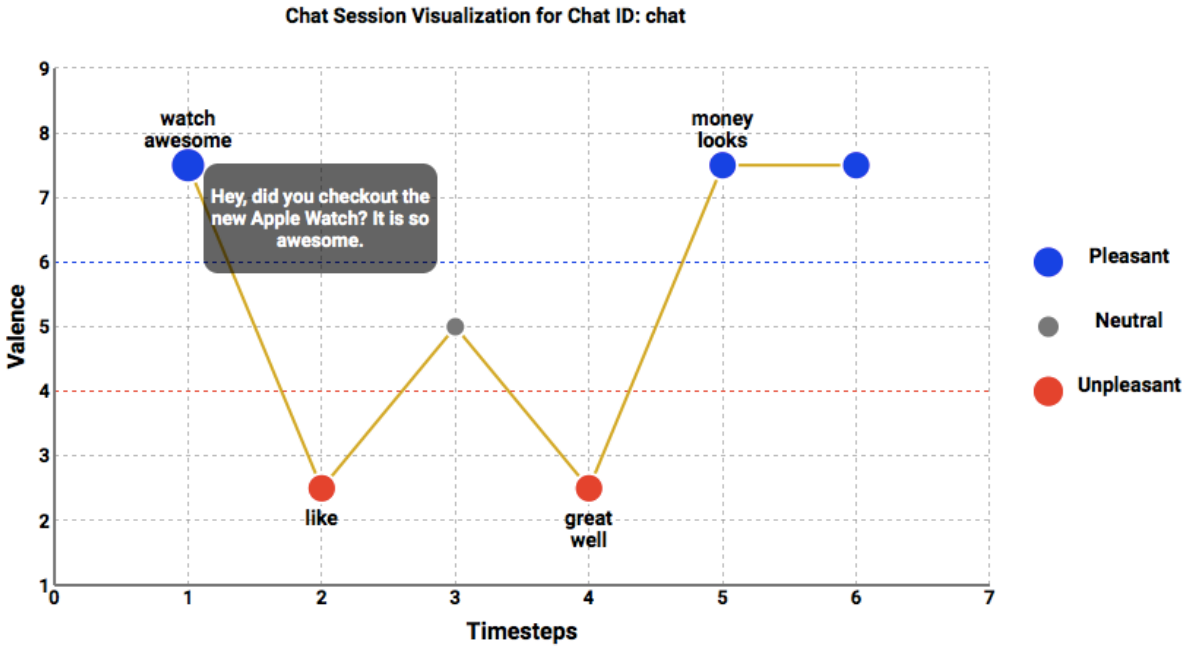

Figure 4.1 Line graph for narrative in the sample chat (The first event is highlighted to show the tooltip).

occurrence in the negative sentiment zone should hint the user that it may be a “not like” occurrence. On further analysis it can be seen that a negation term “didn’t” occurs along with the word “like,” altering its sentiment polarity. Clicking or hovering over an event allows a user to read the entire text within the event. This feature gives the user an option to analyze any event in detail.

Chapter 5

Practical Application

In this chapter we discuss the application of our algorithm to a collection of documents. We collected news articles from the Internet about a specific product, the Apple Watch. We then applied topic clustering and sentiment analysis algorithms proposed in this thesis. We present line graph results for some of the documents within the collection.

5.1 Data Processing

We collected news articles about the Apple Watch product using the search keywords "Apple Watch" from various blogs and product review sites. Text extraction from HTML documents was done using the web scraping tool, Selenium. Figure 5.1 shows the screenshot of a news article from the Internet.

5.1.1 Web Scraping

One significant challenge in collecting data from the Internet was gathering a sufficient number of unique articles about a particular topic. We used online news aggregators, such as *Digg* and *Flipboard*, for this purpose. These sites return links to online articles based on a topic search. The results were web scraped from the respective web pages using Selenium. We retrieved a total of 511 news articles. These articles included product reviews, feature explanations, product release information, and so on. They were written by multiple authors and differed in their textual content.

Unlike other Apple product launches, there will be no long lines snaking around the block on April 24 when Apple Watch becomes available to consumers.

That is because Apple Inc. will only sell them through reservations, according to CNET.

"You won't be able to walk into a store to purchase an Apple Watch like you can with the iPhone and iPad. Instead, all sales will be made through a reservation system," said CNET.

Even trying one on may require some planning. CNET reports that a potential buyer must make an appointment to meet with a store representative for a fitting. Fittings are available starting on Friday, when Apple will start taking preorders for the wearable.

The other option is visiting an Apple store during a downtime and hoping to get lucky.

The iPhone maker is expected to sell as many as 1 million Apple Watches in its opening weekend, putting it on track to sell up to 2.3 million units in the June quarter, according to Piper Jaffray analyst Gene Munster.

Shares of Apple AAPL, +0.12% rose 0.9% to close at $125.32 on Thursday and are up 13.5% this year. Stock markets were closed Friday for the Easter holiday and reopen Monday.

Figure 5.1 A news article on Apple Watch.

5.1.2 Event Generation

Each document was split into paragraphs, with each paragraph in a document acting as an event. This was based on the intuition that each paragraph had related content, and paragraphs were different from each other. For example, an author may talk about the display feature of a product in a paragraph, followed by another paragraph about its battery life. In the sample news article presented in Figure 5.1, the first event is (the first paragraph) is shown in Figure 5.2.

> Unlike other Apple product launches, there will be no long lines snaking around the block on April 24 when Apple Watch becomes available to consumers.

Figure 5.2 The first paragraph of article in Figure 5.1.

In this event, the author expresses his/her view on Apple product launches. While splitting the documents, care was taken to preserve the order of events since they will be merged while structuring the narrative thread for visualization. A total of 5,371 events were extracted from the collection of 511 articles.

5.2 Sentiment Analysis

Dictionary-based sentiment analysis was performed on all events to create a training dataset that was used to train the SAS® SA tool.

5.2.1 Text Pre-Processing

A bag-of-words approach was applied to convert the events into word vectors that were compared to the ANEW dictionary. Each event text input was tokenized to form a set of words by removing punctuation, white space, and other non-text characters, and filtered to remove stop words. Python's Natural Languge ToolKit (nltk) was used in the tokenization process [52].

Table 5.1 Number of events classified under each sentiment.

Sentiment	Number of events classified under the sentiment category
Positive	468
Neutral	4,869
Negative	34

5.2.2 Training Set Generation

Each event was searched for ANEW-recognized words, and sentiment scores were calculated for all the events based on the sentiment estimation method discussed in Chapter 3.1.2. The classification results from the ANEW sentiment analysis are shown in Table 5.1. The results were used as training data for the SA tool.

5.2.3 Document-Level Sentiment Analysis

A SAS® SA model was built by training using ANEW analysis. The ANEW results in each sentiment class (positive, negative, and neutral) were split into 10 different datasets, each with 10% of the total events present in the category. Ten-fold cross-validation was performed for each sentiment class by holding out one of the datasets for testing during each iteration and training using the remaining data. For each iteration of cross-validation, we built a simple statistical model based on the training set.

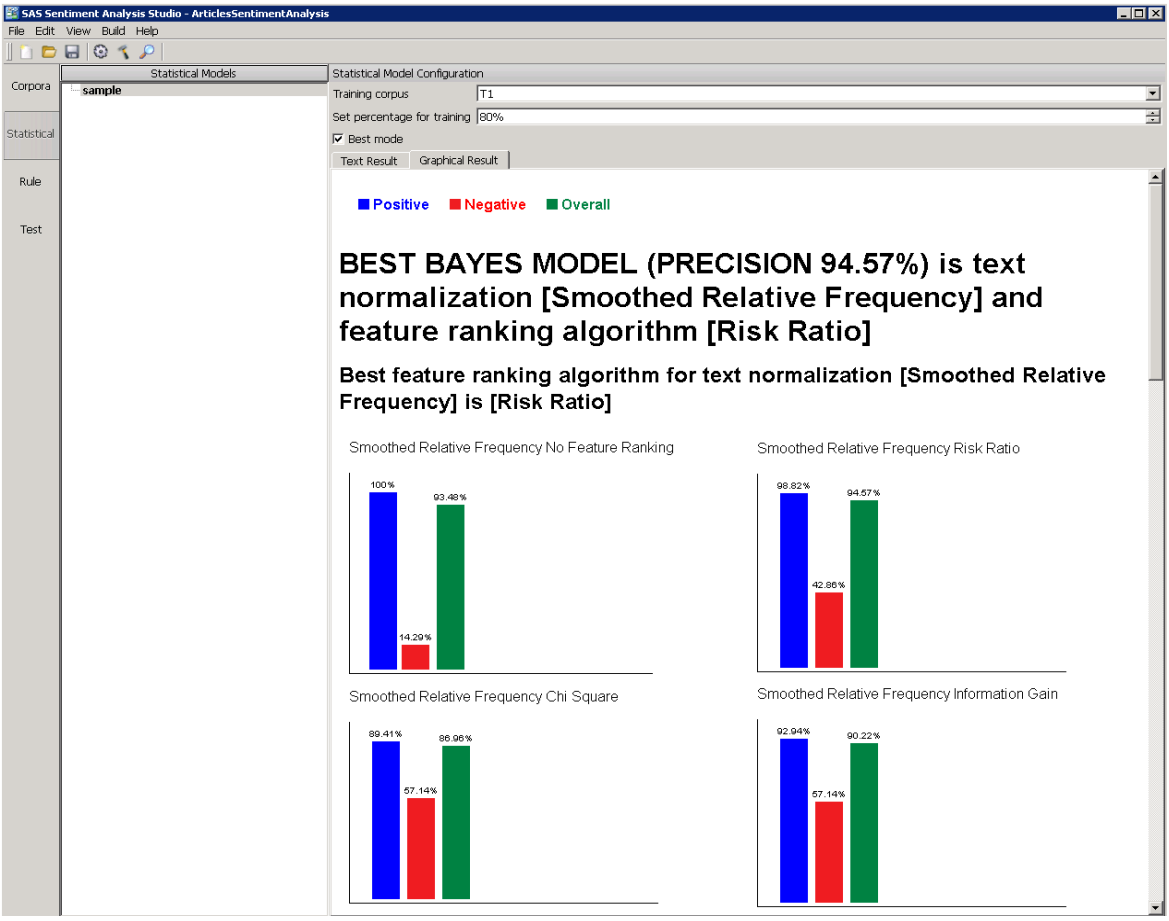

Figure 5.3 A statistical model built in SA using Best mode.

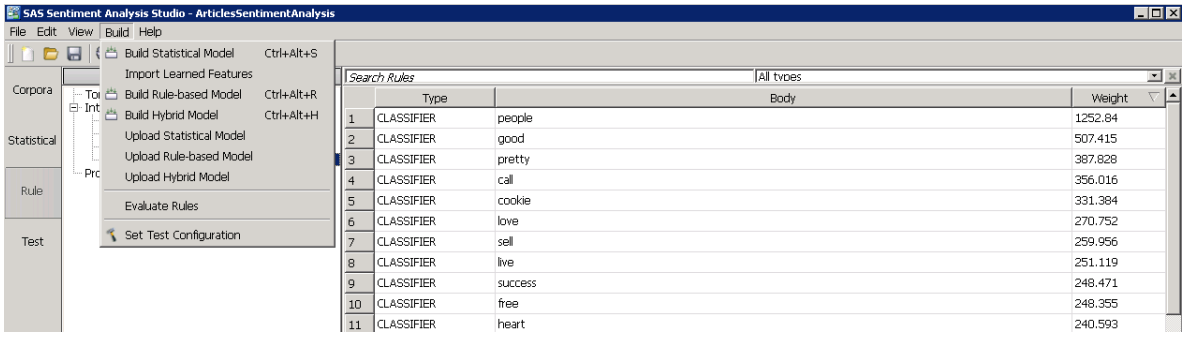

Figure 5.4 Screenshot of the tool showing the option to import rules from statistical model and the positive phrases extracted from the statistical model.

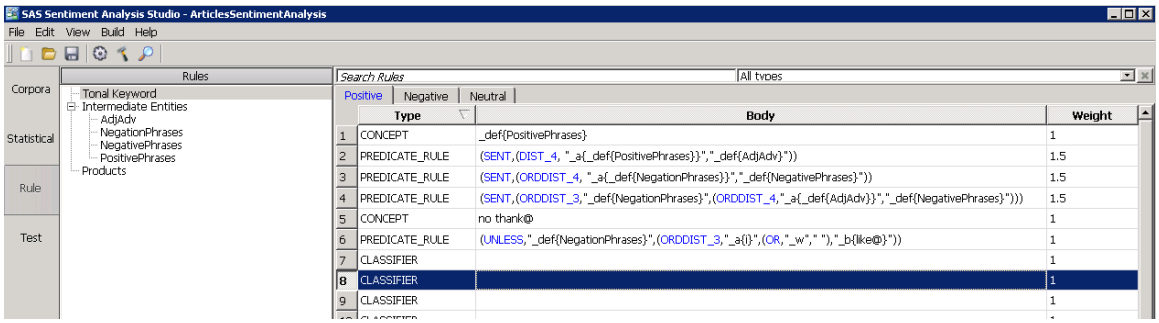

Figure 5.5 Screenshot of the SA tool showing the rules used for matching positive phrases.

Figure 5.3 shows the results from a SA statistical model built from the training set during one iteration of cross-validation. The SA tool has the option of importing keywords from this model, which can be used to construct a rule-based classifier model. Figure 5.4 shows the option of importing keywords ("import learned features") from the statistical model and the phrases that were obtained from the statistical model for positive sentiment. Figure 5.5 shows the keywords categorized into lists under "Intermediate Entities," and the rules written for pattern matching phrases with positive sentiment. The "Negative" tab has the rules for negative sentiment.

After the rule-based classifier model is built, it was run on the test dataset. Figure 5.6 shows a screenshot of the SA tool performing testing on an input directory. One of the test events is misclassified as negative. However, on further observation (Figure 5.7), we see that the event does, in fact, express negative sentiment.

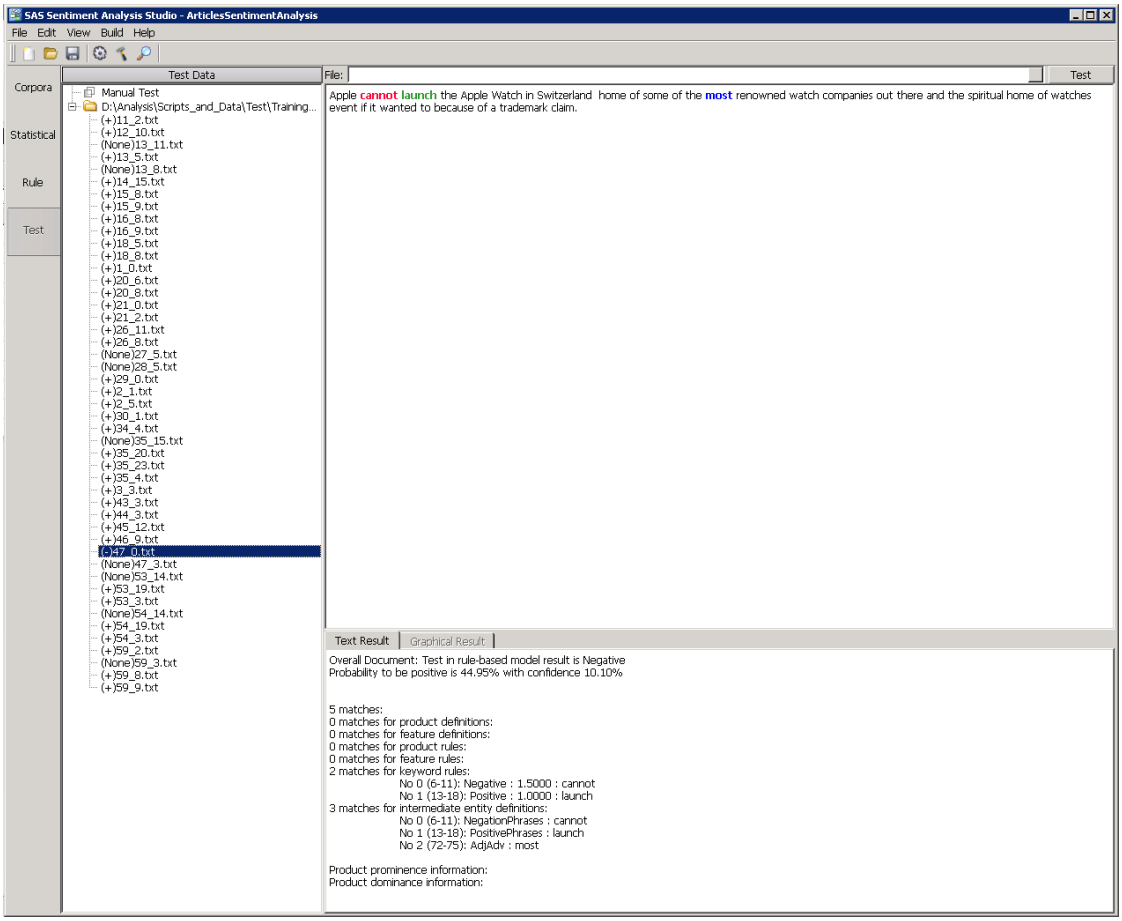

Figure 5.6 Screenshot of results on a test dataset and a misclassification.

Apple cannot launch the Apple Watch in Switzerland, home of some of the most renowned watch companies out there and the spiritual home of watches event if it wanted to because of a trademark claim.

Figure 5.7 Misclassified test event.

The SA identified that “launch,” which is in the positive phrases list, is negated by the word “cannot,” which is in the negation phrases list, and hence, detected by the negation rule under negative pattern matching, which has a greater weight. Thus, the event is correctly classified as negative by SA. ANEW analysis is not intelligent enough to detect such patterns.

In the opposite case where events were classified correctly by ANEW but incorrectly by the SA model, the rules causing this issue were identified and fixed. Thus, the process of refining the model involves manual verification of the test data set, which was classified by ANEW sentiment analysis during training set generation, with corrections to the SA rules as necessary. This was repeated until the accuracy of the model reached at least 75% for each test dataset. Finally, phrases and rules from all the models were merged to create a single model.

The final SA model was used to classify all the events in the document collection. Table 5.2 shows the events’ classification results. There is a significant rise in the number of events classified under positive and negative, when compared to the results from ANEW analysis. This is because statistical modeling using the SA tool helped discover more domain specific words that express sentiment, which were not detected by ANEW analysis.

Table 5.2 Sentiment classification results from SAS® SA tool.

Sentiment	Number of events classified under the sentiment category
Positive	3217
Neutral	1848
Negative	306

Table 5.3 Number of documents under each sentiment category.

Sentiment	Number of documents classified under the sentiment category
Positive	451
Neutral	30
Negative	30

The document-level sentiment is estimated by finding the dominant sentiment of all the events within it, as described in Section 3.3. The approach considers both the confidence of sentiment ratings and the number of events within the document. Documents may not be positive even when they contain more positive events than negative events, because the overall confidence rating of the positive events may be less than the confidence rating of the negative events. In this case, the document is given a negative sentiment. Table 5.3 shows the classification results of the collection of documents.

Apple cannot launch the Apple Watch in Switzerland - home of some of the most renowned watch companies out there and the spiritual home of watches - event if it wanted to because of a trademark claim.

A company called Leonard Timepieces filed a trademark claim in 1985, which essentially bars Apple and any other company from using the word apple or the image of an apple for selling precious metals and their alloys and goods in these materials or coated therewith, not included in other classes; jewelry, precious stones; clocks and timepieces.

The trademark was filed for 30 years and expires on December 5, 2015. Apple can easily wait and launch the Apple Watch in Switzerland after that, but the Cupertino company is apparently in talks with the owner of Leonard Timepieces, William Leong, to negotiate a deal and purchase the trademark.

It is also entirely possible that Leong renews the trademark instead of selling it to Apple to the benefit of Swiss watchmakers.

The Apple Watch is all set to go on pre-orders on April 10 in nine countries, including the United Kingdom, United States, France, Germany and Hong Kong.

Figure 5.8 Article No. 47.

5.3 Narrative Visualization

Narrative threads were constructed based on the sentiment results of events within the documents. A web application was developed to view each document individually. Figure 5.9 shows the narrative visualization of one of the news articles in the collection. Table 5.4 shows the sentiment results of the events within that article. It can be observed that the overall confidence rating of the positive events is greater than that of the negative events, and hence, the document is classified as positive. The article text is shown in Figure 5.8.

Table 5.4 Sentiment results of all events in article No. 47.

Event ID	Sentiment	Confidence Rating
1	Negative	0.101
2	Neutral	0
3	Positive	0.2
4	Neutral	0
5	Positive	0.2

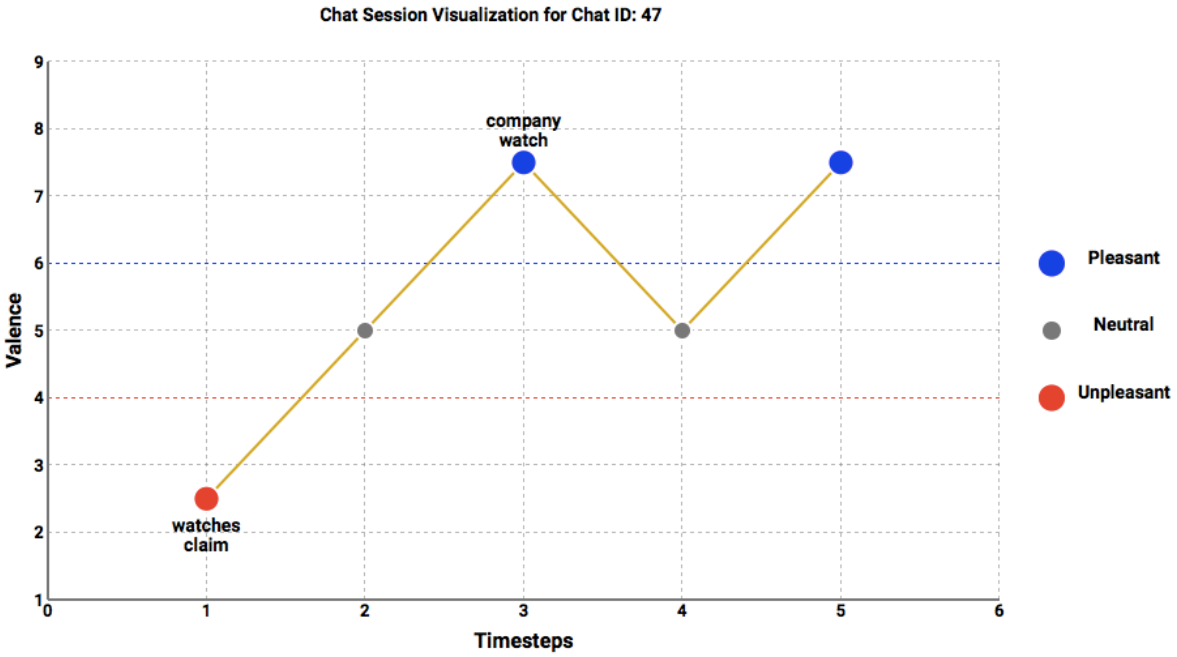

Figure 5.9 Narrative thread visualization of article No. 47 within the collection.

The article explains the problem faced by Apple in their release of Apple Watch in Switzerland due to a trademark claim made by another company, which prevents Apple from

using their brand name “Apple” in the selling of any of their products. The first paragraph (event) states the problem, and is classified as negative. The third event describes how Apple can “easily” overcome the problem, which expresses a positive sentiment, and thus is classified positive. This sentiment transition is identified and corresponding events are tagged with ANEW-recognized keywords within the paragraph. The keywords in the first event convey the idea that the negative sentiment is due to some “claim” related to the term “watch.” Similarly, the third event shows that there is some positive news for the “company” and the “watch.”

Figure 5.10 shows an example of an article, which was classified as negative by SA. The sentiment of events (paragraphs) within the document is shown in Table 5.5. The article discusses the security issues that the Apple products are facing. It starts with excitement about the new products introduced by Apple and goes on to describe the lack of stronger security features. Towards the end of the article, the author is concerned about whether Apple will neglect enhancing security for its new payment feature, which will put more user data at risk, and how dangerous that would be. But, the article ends with a hope that the company reviews its security policies.

It can be observed from Table 5.5 that there are more events with positive sentiment than negative sentiment. But, using the algorithm described in Section 3.3 for estimating document-level sentiment, we find the average confidence of positive sentiment to be 0.3213 and the average confidence of negative sentiment to be 0.8978. Hence, the document is classified as negative, since the negative sentiment of the events within it dominate. Figure 5.11 shows the narrative thread visualization of the article.

Today's event offered Apple fans a lot to get excited about: a new kind of watch, a huge iPhone and a new way to pay for things, for a start. ... Just a week after attackers penetrated iCloud, stealing private photos from the services most famous users, Apple made no reference to new security features, or plans to lock down its massive infrastructure.

On some level, this isn't surprising. The iPhone 6 was the star of the show, a show Apple has been planning for years, so it's understandable if Tim Cook didn't want to spoil the new iPhone's debut with references to this month's bad PR. ... Unfortunately for Apple, it's a question that's getting harder and harder to answer.

On some level, Apple Pay has a lot going for it. ... Thanks to TouchID, the system even has fingerprint-based authentication on its side. But its most important security feature is actually a strange admission of failure: Apple Pay appears to work completely separately from iCloud. Healthkit has the same feature, warning developers upfront not to store any sensitive health data on iCloud. It leaves Apple users in a strange place, rushing into a new infrastructure while increasingly uncertain about the security of the old one. ...

With Apple Pay, Apple will be tackling a whole new set of problems, and it's easy to be concerned that the payments project will succumb to the same neglect as iCloud. To be clear, the problem isn't that Apple is particularly bad at security. ... Apple isn't any worse at bug-hunting than its competitors. It's just farther ahead on everything else. Apple Pay puts more data at risk and offers more ways to get at it. That's a dangerous combination if security isn't keeping pace.

Maybe I'm wrong. Maybe the latest iCloud problems really were a wakeup call, as Cook has suggested, and the company is reviewing its protocols behind the scenes. ... I hope it happens. But on the heels of Apple's biggest announcement in years, I'm more worried than ever.

Figure 5.10 Article No. 369

Table 5.5 Sentiment results of all events in article No. 369.

Event ID	Sentiment	Confidence Rating
1	Positive	0.4675
2	Positive	0.5429
3	Negative	0.9898
4	Positive	0.2951
5	Positive	0.2
6	Negative	0.8058
7	Positive	0.1010

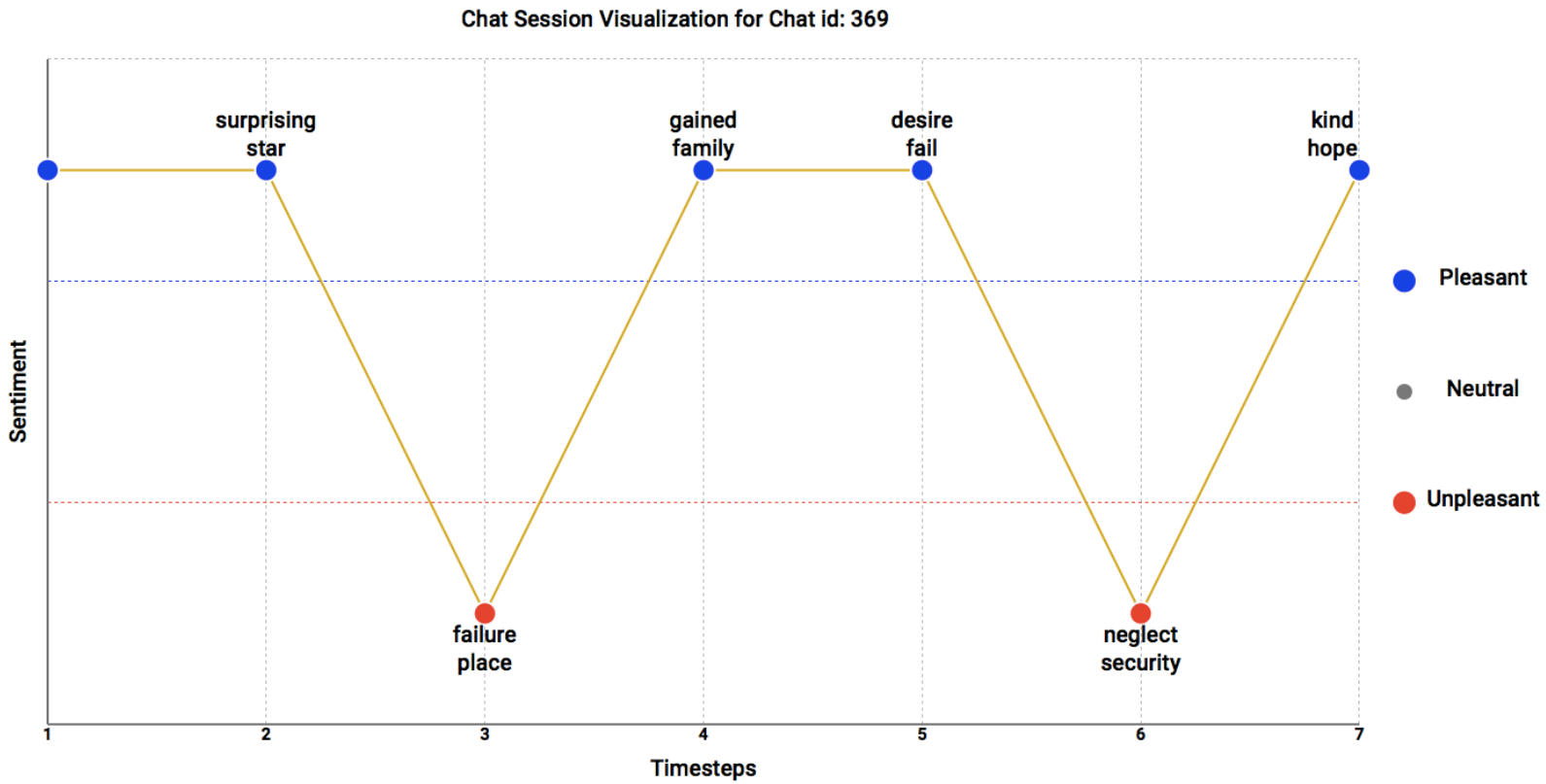

Figure 5.11 Narrative thread visualization of article No. 369 within the collection.

Apple is offering employees a 50 percent discount on the Apple Watch in the hope it becomes a fixture on the wrist of everyone who gets a pay cheque signed by Tim Cook.

In a letter sent out to staff by the CEO himself, Cook said the troops can take advantage of buying a half price watch as soon as pre-orders kick off on Friday.

Naturally, the offer doesn't quite stretch to the Apple Watch Edition, thus Cupertinos finest won't be eligible for a discount worth thousands of dollars.

Instead, those intent on strapping on the 18-carat gold smartwatch will get up to $550 (around €367) off. Staffers will have 90 days to claim their discount.

In the memo (via 9to5Mac), Cook wrote: I know that many of you have been looking forward to choosing an Apple Watch for yourselves, and we want to make it easy for you. ...

Our products enrich peoples lives like no others and we think Apple Watch is going to delight our customers in ways people can't yet imagine. We want you to share in that experience alongside them.

The deal is a boost for the staff, who will surely be expected to drop their trusty wrist watch in favour of the company's own interpretation.

It'll be especially important for those Apple employees who don the blue sweaters at Apples retail outlets to be seen representing.

In terms of discounts, it has been a mixed bag for Apple employees down the years. ... However they were forced to wait before becoming eligible for a discount for an iPad when it launched in 2010.

Elsewhere in the memo, Cook revealed 1,000 apps with Apple Watch support have been submitted to the App Store since the company opened the floodgates late last week.

Apple Watch owners should have plenty to choose from when the watch actually goes on sale on April 24.

Figure 5.12 Article No. 13

Figure 5.12 shows an article, which was classified as positive and had no negative events. This article is about the discount on Apple products offered to Apple employees. Thus, the events within the articles are positive in general, apart from some, which are neutral. None of the events were negative. Therefore, the document is classified as positive. The narrative thread within the article is shown in the Figure 5.13. Table 5.6 shows the sentiment of events within the article.

Table 5.6 Sentiment results of all events in article No. 15.

Event ID	Sentiment	Confidence Rating
1	Positive	0.2
2	Neutral	0
3	Neutral	0
4	Positive	0.3846
5	Positive	0.7673
6	Positive	0.3846
7	Neutral	0
8	Neutral	0
9	Positive	0.4675
10	Positive	0.2
11	Positive	0.2

5.4 Performance Versus Stand-Alone SA

We analyzed the performance of our algorithm by comparing it to a SA model alone. We used a collection of documents from a chat domain, which contained about 50,000 chat transcripts between a company's customer care representatives and consumers. Approximately ten percent of the chat documents had feedback data from the consumers, rating the quality of the conversation on a scale of one to five. We classified the sentiment of chats with feedback based on their ratings. We then used the results as a training set to create a SA model, which automatically classified new text input. We then applied the algorithm that we presented in this thesis by using ANEW-based sentiment analysis to create a training set, and then a SA model. Each chat document was split into events, and sentiment was estimated for individual events. Document sentiment was estimated based on the dominant sentiment among the events within it.

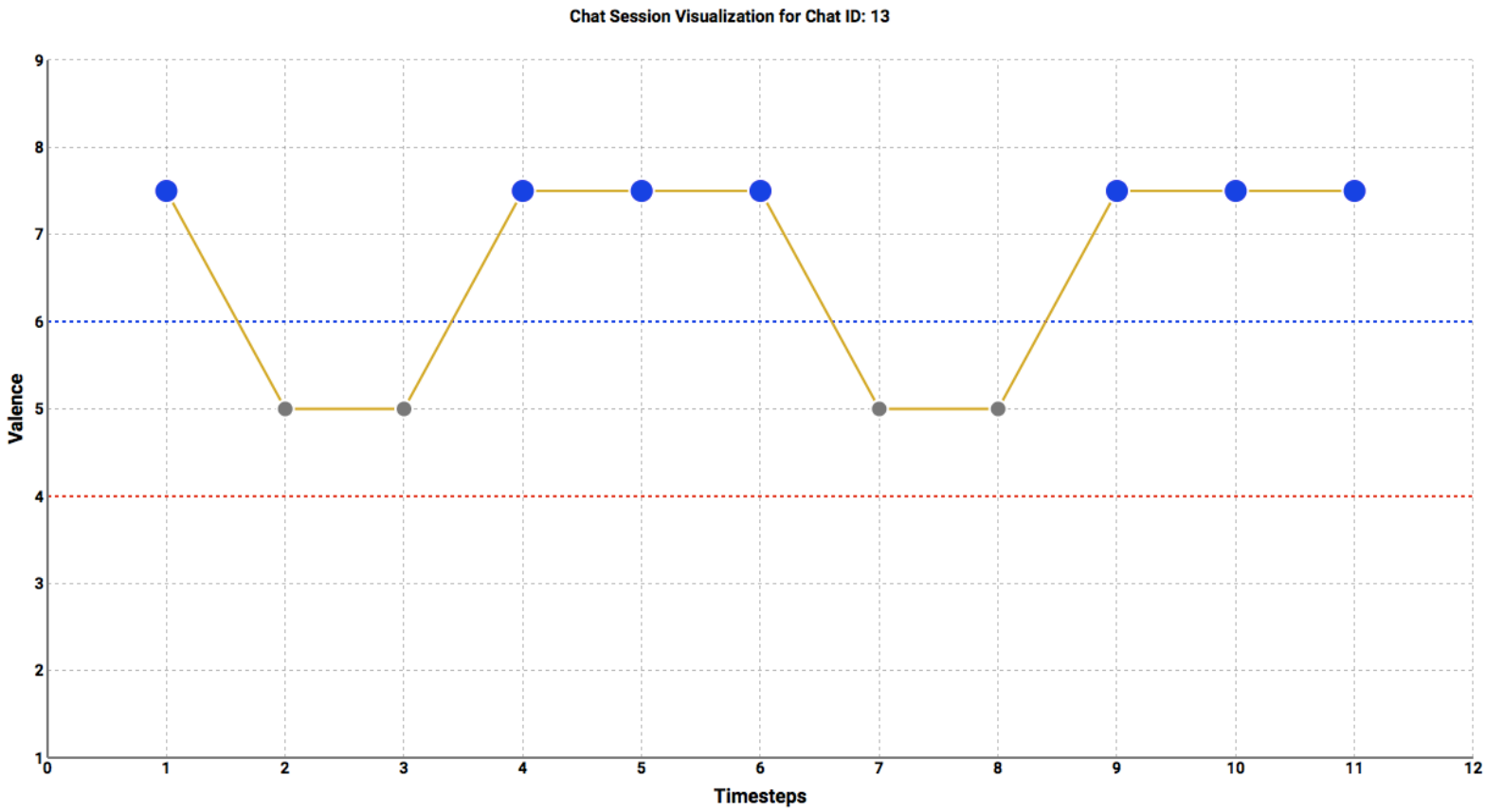

Figure 5.13 Narrative thread visualization of article No. 369 within the collection.

The results obtained from our approach were an improvement over using the SA model alone, and also gave more information about the sentiment transitions within a document. We hypothesize the following factors enabled our algorithm to perform better:

1. Event-based approach. We split each document into events and analyzed their sentiment instead of considering the document in its entirety.
2. Dictionary-based training set generation. Generating a training set based on a sentiment dictionary helps detect words that express sentiment, which is further improved by using the rule-based classifier in SA.

Chapter 6

Conclusions and Future Work

Sentiment analysis is evolving from a research area into a practical text analysis task. Growth of user-generated content has made sentiment analysis a useful application in determining the overall feeling of consumers towards a product or a topic. Many companies want to use sentiment analysis to understand their users' mood towards their brand in order to provide better service. The major challenge in sentiment analysis is the generation of hand-classified training data sets, which are necessary to build the statistical models that automatically classify input documents. We have successfully implemented an effective method for automatically generating training data set for sentiment analysis, which is not restricted to any particular domain. Dictionary-based sentiment analysis is used to generate the initial training data sets. We used the ANEW dictionary for this task although any sentiment dictionary with valence scores could be used. While the users still need to verify the results from ANEW when building the model using a follow-on tool, the time spent on validation is much less than fully hand-classifying a sufficient number of events to form a training set. We estimate the document-level sentiment by first splitting the documents into events, analyzing sentiment on each of these events, and finally merging their sentiment results. This gives a more detailed and more accurate estimate of document-level sentiment, versus analyzing the document in its entirety.

We structured narrative threads from text documents based on their sentiment results and used line graphs to visualize them. Critical events were identified, and keywords within

these events were tagged to give the users clues about the sentiment transition. We tested our approach using 511 online news articles about Apple Watch. The sentiment results obtained were good for both documents and events within the documents. Building narrative structure using sentiment conveyed significant information about the documents, all of which was visually represented in our line graph. We were able to identify problems or dissatisfaction about the product from the events with negative sentiment, and solutions or positive factors from the events with positive sentiment. Appropriate colors were used to represent sentiment to clearly distinguish different sentiments and emphasize positive and negative events.

The sentiment analysis technique works by dividing a document into events. When a document has very large number of events, particularly with many sentiment transitions, the line graph visualization becomes crowded. This may obstruct the observation of critical events. One way to overcome this would be to fix the horizontal scale of line graph and make the visualization scrollable. Although the validation data source used was online news articles, the technique was found to be effective when applied to a chat data set. This suggests the technique should also work well for social media data, since they have a strong cause and effect relationship between events, similar to chat data and, possibly more so compared to the news articles. Further validation would be needed to confirm this hypothesis.

Although, the standard deviation rating provided by the ANEW dictionary expresses negation to some extent, it does not explicitly handle negation of sentiment. Future work could implement negation-handling techniques in the dictionary-based sentiment analysis for

constructing training sets, which could be effective in reducing the time spent in the process of correcting the training set results.

Finally, the current line graph visualization works only for individual documents. Future work may try to group documents based on the number of sentiment transitions within them and present an overall view of the text corpus.

References

[1] Campbell, J. J. (2008). *The Hero with a Thousand Faces* (p. 30). New World Library.

[2] Munroe, R. (2009). Movie Narrative Charts. *Xkcd*. Retrieved from https://xkcd.com/657/

[3] James, J. (2014, April). Data Never Sleeps 2.0. Retrieved from https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/

[4] Rockwell, G. (2001). The Visual Concordance: The Design of Eye-ConTact. In *Text Technology* (pp. 73-86).

[5] Holmes, D. I. (1998). The Evolution of Stylometry in Humanities and Scholarship. *Literary and Linguistic Computing, 13*(3)*,* 111-117.

[6] Williams, C. B. (1976). Mendenhall's studies of word-length distribution in the works of Shakespeare and Bacon. *Biometrika, 62*, 207–212.

[7] Hearst, M. A. (1996). Untangling Data Mining. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics,* 3–10.

[8] Hotho, A., Nürnberger, A., and Paaß, G. (2005). A Brief Survey of Text Mining. *Journal for Computational Linguistics and Language Technology, 20,* 6-19.

[9] An algorithm for suffix stripping. (1997). In *Readings in information retrieval* (pp. 313–316). Morgan Kaufmann Publishers Inc.

[10] Andrews, K. (1971). The Development of a Fast Conflation Algorithm for English (Unpublished doctoral dissertation). University of Cambridge.

[11] Manning, C. D., Raghavan, P., & Schutze, H. (2009). The term vocabulary and postings lists. In *Introduction to Information Retrieval* (pp. 33–34). Cambridge University Press. Retrieved from http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

[12] Salton G., Wang, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

[13] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, *25,* 259-284.

[14] Maron, M. E., & Kuhns, J. L. (1960). On Relevance, Probabilistic Indexing and Information Retrieval. Journal of the ACM (JACM), 7(3), 216–244.

[15] Borko, H., & Bernick, M. (1963). Automatic Document Classification. *Journal of the ACM (JACM)*, *10*(2), 151–162.

[16] Steinbach , M., Karypis , G., & Kumar, V. (2000). Automatic Document Classification. *KDD Workshop on Text Mining*.

[17] Vogt, C., & Alldredge, K. (2012). Understanding the Role of the Internet in the Lives of the Consumers. *2012 Digital Influence Annual Global Study*. Retrieved from http://www.harrisinteractive.com/vault/HI_UK_Corp_Insights-Fleishman-Hillard-DDI-2012.pdf

[18] Wiebe, J., & Bruce, R. (1995). Probabilistic classifiers for tracking point of view. *Proceedings of the AAAI Spring Symposium on Empirical Methods in Dis- Course Interpretation and Generation*, 181–187.

[19] Liu, B. (2012). *Probabilistic classifiers for tracking point of view*. Morgan & Claypool Publishers.

[20] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceeding EMNLP '02 Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, *10*, 79–86.

[21] May, T. (2002). Narrative in Social Research. In *Qualitative Research in Action* (pp. 242–243). SAGE.

[22] Tufte, E. (1983). In *The Visual Display of Quantitative Information*. Graphics Press.

[23] Census Data Mapper. (2010). Retrieved from http://tigerweb.geo.census.gov/datamapper/map.html

[24] Healey, C. (2014). 2014 U.S. Election Visualizations. Retrieved from http://www.csc.ncsu.edu/faculty/healey/US_election/

[25] Friendly, M. (2005). Milestones in the History of Data Visualization: A Case Study in Statistical Historiography. In *Classification: The Ubiquitous Challenge* (pp. 34–52). Springer.

[26] New York City Department of Transportation. (2008). *Safe Streets NYC: Traffic Safety Improvements in New York City.* New York, NYC DOT Library.

[27] Google Finance. (2015). Retrieved from http://www.google.com/finance?q=INRUSD

[28] Tan, P., Steinbach, M., & Kumar, V. (2006). Visualization. In *An Introduction to Data Mining* (pp. 97–118). Pearson.

[29] Anderson's Iris Data Set. (2007). Retrieved from http://en.wikipedia.org/wiki/File:Anderson%27s_Iris_data_set.png

[30] Polti, G. (1916). *Thirty-Six Dramatic Situations*. Franklin, O., J.K. Reeve.

[31] Wojtkowski, W., & Wojtkowski, G. W. (2002). Storytelling: its role in information visualization. *European Systems Science Congress*.

[32] Havre, S., Hetzler, B., & Nowell, L. (2000) ThemeRiver: Visualizing Theme Changes over Time. In *Proceedings of the IEEE Symposium on Information Visualization*, 115–123.

[33] Tanahashi, Y., & Ma, K.-L. (2012). Design Considerations for Optimizing Storyline Visualizations. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, *18*(12), 2679–2688.

[34] Healey, C., & Ramaswamy, S. (2011). Visualizing Twitter Sentiment. Retrieved from http://www.csc.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

[35] Bilenko, N., & Miyakawa, A. (n.d.). Visualization of Narrative Structure. Retrieved from http://nbilenko.com/projects/narrative.html

[36] Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, *11*(1), 92–99.

[37] Quealy, K., & McClain, D. L. (2015, September 15). A Year of Heavy Losses. *New York Times*. Retrieved from http://www.nytimes.com/interactive/2008/09/15/business/20080916-treemap graphic.html?_r=1&

[38] Feinberg, J. Wordle. In *Beautiful Visualization: Looking at Data through the Eyes of Experts (Theory in Practice)*, O'Reilly Media, 2010, (pp. 37–58).

[39] Wikipedia. (2001). Retrieved from http://en.wikipedia.org/wiki/Main_Page

[40] Esuli, A., and Fabrizio S. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC),* 417–422.

[41] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. A Bradford Book.

[42] Bradley, M.M., & Lang, P.J. (1999). Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical report C-1, Gainesville, FL. The Center for Research in Psychophysiology, University of Florida.

[43] Warriner, A. B., Kuperman, V., & Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 English lemmas. In *Behavior Research Methods, 45,* 1191-1207.

[44] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. In *Journal of the American Society for Information Science and Technology, 61*(12), 2544-2558.

[45] McNair, D. M., Lorr, M., & Droppleman, L. F. (1971). Manual for the Profile of Mood States. San Diego, CA: Educational and Industrial Testing Services.

[46] Bollen, J., Pepe, A., & Mao, H. (2009). Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena. In *CoRR Technical Report.*

[47] Bollen, J., Pepe, A., & Mao, H. (2011). Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM11)*.

[48] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 168–177.

[49] Dodds P. S., Danforth C. M. (2009). Measuring the happiness of large-scale written expression: songs, blogs, and presidents. In *Journal of Happiness Studies 11*(4), 441-456.

[50] Gutierrez-Osuna, R. (n.d.). Validation. Retrieved from http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf

[51] Reckman, H., Baird, C., Crawford, J., Cowell, R., Micciulla, L., Sethi, S., & Veres, F. (2013). Rule-based detection of sentiment phrases using SAS® Sentiment Analysis. *SemEval-2013 Task 2, A and B*.

[52] Natural Language ToolKit. Retrieved from http://www.nltk.org/index.html