Plot Extraction and the Visualization of Narrative Flow

Michael A. DeBuse
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Physical Sciences and Mathematics Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/9666

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Plot Extraction and the Visualization of Narrative Flow

Michael A. DeBuse

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Sean Warnick, Chair
Deryle Lonsdale
David Wingate

Department of Computer Science
Brigham Young University

Copyright © 2021 Michael A. DeBuse
All Rights Reserved
ABSTRACT

Plot Extraction and the Visualization of Narrative Flow

Michael A. DeBuse
Department of Computer Science, BYU
Master of Science

In order to facilitate the automated extraction of complex features and structures within narrative, namely plot in this study, two proof-of-concept methods of narrative visualization are presented with the goal of representing the plot of the narrative. Plot is defined to give a basis for quality assessment and comparison. The first visualization presented is a scatter-plot of entities within the story, but due to failing to uphold the definition of plot, in-depth analysis is not performed. The second visualization presented is a graph structure that better represents a mapping of the plot of the story. Narrative structures commonly found within the plot maps are shown and discussed, and comparisons with ground-truth plot maps are made, showing that this method of visualization represents the plot and narrative flow of the stories.

Keywords: natural language processing, computational linguistics, digital humanities, narrative flow, content extraction, plot, narrative visualization
ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Sean Warnick, for his guidance and assistance in this research. I would also like to thank Dr. Deryle Lonsdale for his instruction that inspired me to first begin my research with the creation of the Scatter-plot of Entities. Lastly I would like to give a special thanks to Alyssa Crezee, Caelen Miller, Maxwell Hamilton, Xin Zhou, Henry Ward, and Griffin Holt of my annotation team for their assistance with data preparation and creation of the ground truth data.
# Table of Contents

Title Page i

Abstract ii

List of Figures v

1 In Preparation: Plot Extraction and the Visualization of Narrative Flow 1

References 27
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2</td>
<td>7</td>
</tr>
<tr>
<td>Figure 3</td>
<td>8</td>
</tr>
<tr>
<td>Figure 4</td>
<td>8</td>
</tr>
<tr>
<td>Figure 5</td>
<td>14</td>
</tr>
<tr>
<td>Figure 6</td>
<td>17</td>
</tr>
<tr>
<td>Figure 7</td>
<td>18</td>
</tr>
<tr>
<td>Figure 8</td>
<td>19</td>
</tr>
<tr>
<td>Figure 9</td>
<td>19</td>
</tr>
<tr>
<td>Figure 10</td>
<td>20</td>
</tr>
<tr>
<td>Figure 11</td>
<td>20</td>
</tr>
<tr>
<td>Figure 12</td>
<td>20</td>
</tr>
<tr>
<td>Figure 13</td>
<td>22</td>
</tr>
<tr>
<td>Figure 14</td>
<td>23</td>
</tr>
<tr>
<td>Figure 15</td>
<td>23</td>
</tr>
</tbody>
</table>
Chapter 1

In Preparation: Plot Extraction and the Visualization of Narrative Flow

This manuscript has not yet been accepted for publication.

The entire body of this thesis document is contained within this manuscript, including introduction, apposite research, methods, results, discussion, and references.
Plot Extraction and the Visualization of Narrative Flow

Michael A. DeBuse*
Brigham Young University

In order to facilitate the automated extraction of complex features and structures within narrative, namely plot in this study, two proof-of-concept methods of narrative visualization are presented with the goal of representing the plot of the narrative. Plot is defined to give a basis for quality assessment and comparison. The first visualization presented is a scatter-plot of entities within the story, where clustering around entity density is used to partition scenes. The second visualization presented is a graph structure of the narrative flow that better represents a possible mapping of the plot of the story. Narrative structures commonly found within the graphs are shown and discussed, and comparisons with ground-truth narrative flow graphs are made.

1. Introduction

In Franco Moretti’s influential book, Distant Reading (Moretti 2013), he claims that in order for us to better understand literature as a whole, we should change our focus to information extraction tools to aid us in analyzing the content of literature instead of simply reading it, a stance that has received both support and criticism among the academic community. The reason is simply that there are more published books than any one person or small group of people could ever read. According to Google’s book search algorithm, there are about 130 million books that have been published as of 2010 (Taycher 2010). There are many natural language processing (NLP) tools that aid in the evaluation of text, such as dependency parsing, named entity recognition, and sentiment analysis, but there is a lack of readily available tools that analyze or extract complex aspects of literature, such as plot and narrative flow.

Plot can be described as the causal interaction of key elements and events in a story that progress the story from its beginning to its end (a more rigorous definition will follow in Section 2.1). Currently, in order to do any plot analysis of a novel, news story, historical account, or any other form of narrative where plot is present, the text must be read by a human who then performs the analysis. For a common-length novel of about 100,000 words and an average reading speed of 200 words a minute, one novel can take over eight hours to read. That time requirement is then multiplied by the number of books that need to be read and is then compounded by the time required to perform the analysis. The purpose of this research is to drastically reduce this time by automating the extraction of the plot from text where plot exists, such as novels, and output it in a form that is both easy to understand visually and also usable in other analytical software and machine learning systems. Although this research can be applicable to other media where a narrative exists, such as news stories and historical accounts, for this research I focus on fiction due to the rich and diverse plots the medium makes possible.

* E-mail: mdebuse3@gmail.com
In this research I present two visualizations of the narrative of the story. The **Scatter-plot of Entities** visualizes the introduction of entities (actors, objects, locations, etc.) within the story and arrays them on a scatter-plot according to their continued appearance in narrative order. I then attempt to divide the narrative into scenes using clustering. The **Plot Map System** uses the story entities and locations to build a graph structure representing the narrative flow of the story. Scenes act as vertices of the graph, and the entities become the edges connecting the scenes from the beginning of the story until the end.

2. History and Related Work

2.1 On Plot and Narrative

The description of plot given in the introduction is insufficient for complex analysis and extraction, namely because it does not detail what needs to be extracted. The interacting elements that progress the story from beginning to end must be defined.

Plot and narrative structure are closely linked. Plot drives the story while narrative is how the story is told, often called narrative discourse or simply the discourse (Puckett 2016). The link between plot and the structure of the narrative is so inherent that when many think of plot, they think of structures or steps, almost algorithmic, that are laid in order. An example of such a structure is as follows: exposition, inciting incident, rising action, climax, falling action, denouement, resolution. This example is commonly called Freytag’s Pyramid and was used to describe the narrative structure of classical epics and dramas (Freytag 1863). However, not all stories follow the same structure. In addition to Freytag’s Pyramid, there is the Fichtean Curve, Hero’s Journey, In Media Res, 3-act, Seven Point, and more. If we are to consider this type of structure labeling as necessary for plot or to be plot itself, there would need to be a universal structure, or some way to define a structure that then could be applied universally throughout.

Formalist literary researchers and critics break down the structure to smaller thematic elements. Vladimir Propp delved into Russian folktales to investigate the commonalities between them (Propp 1968). His idea was to separate the theory from the specifics and assign labels to the different forms the events and characters in a story can take, resulting in 31 narratimes. Boris Tomashevsky wrote of the microstructure of a story, how it can be broken down into what are very similar to Propp’s narratimes, thematic sections of the story, or events, that follow specific forms (Tomashevsky et al. 1965). Alexander Veselovsky spoke of the motif, or the “simplest narrative unit” of a story, that combines together to create the themes of a tale (Veselovsky 2015). In an expansion on Propp and Veselovsky and inspired by other folklorists like Antti Aarne, American folklorist Stith Thompson developed the Motif-Index of Folk-Literature, 6 volumes that include thousands of commonly occurring—and some rarely occurring—event types and story elements in a folktale (Thompson 1989). It is clear to see that as more investigation is done into thematic elements of story, the number of those elements ever increases. Unless the scope is narrowed, identifying every thematic element in a story is a monumental task, so the structure must be broken down further.

At its core, plot involves action. In part 6 of the Poetics (Aristotle 1961), Aristotle claims that one cannot have plot without action. This notion comes from the Tragedies and other stage plays of the period, where visible action is needed to understand the plot, and the lack of action means the absence of plot. The Aristotelian notion of action-driven story has held for many centuries, but it alone is insufficient to represent the complexities of plot, especially in literature. E. M. Forster theorizes that plot is more
than the Aristotelian notion of action-driven story. He states that what is known and
not known as well as the emotions that lead to action are just as vital as the actions
themselves. In addition, the causal element is at the core of plot. In a famous example,
Forster states, ‘A plot is also a narrative of events, the emphasis falling on causality.
The king died and then the queen died,’ is a story. ‘The king died, and then the queen
died of grief’ is a plot’ (Forster 1927). In The Plot of Tom Jones (Crane 1950), R. S. Crane
elaborates on Forster’s idea and criticizes Aristotle, defining plot as the synthesis of
action, character, and thought that may take on different structures depending on the
author’s use and emphasis on any of these three aspects.
Plot, therefore, requires five elements:

• **Characters:** entities of volition within a story
• **Events:** actions taken in the story
• **Information:** what is known and how that knowledge spreads
• **Causation:** the manner in which events are linked, one leading to another
• **Structure:** the complete linking of events from the beginning of the story
to the end

The output created by the two extraction methods detailed in this paper are addi-
tionally analyzed on how they fulfill this definition of plot.
It is also important to note here that when narrative is mentioned in this paper,
it refers to the the source text and how that text represents the underlying story (the
discourse). With the focus of this research on plot and narrative structure, *narrative flow*
is then how the events and scenes in that narrative progress from one to the next. This
is different than the flow of how the narrative sounds and feels when read or spoken.

### 2.2 Relation to Event Extraction

Aristotelian action-driven story may be an insufficient definition to fully encapsulate
the complexity of plot, but physical actions are still a major aspect of plot, and for most
novels it is the dominant element. This means that plot extraction can be seen as a form
of event extraction modified to fit the definition of plot.

Little research has been made into applying event extraction to fiction text. Much
of the research involves biomedical (Yakushiji et al. 2001; Riedel and McCallum 2011;
Bjorne and Salakoski 2011), news (Naughton, Kushmerick, and Carthy 2006; Vargas-
Vera and Celjuska 2004), and historical text (Chieu and Lee 2004; Segers et al. 2011), to
name a few. Each of these areas and more require a more topical approach to event ex-
traction where specific trigger or anchor words commonly found in the desired topical
domain help identify the events of the text to be extracted. This is not feasible for fiction
literature due to the plethora of topics, themes, and genres. Non-topical or domain-free
event extraction has been used in studies like those of Alan Ritter et al. to extract events
of general interest from Twitter using machine learning to recognize event structure
(Ritter, Etzioni, and Clark 2012) and Valenzuela-Escarcega et al. in biomedical text using
rule-based algorithms that detect events by locating sentences that have the correct
grammatical or semantic representation of the desired event (Valenzuela-Escarcega et al.
2015). This removes the need for trigger words and allows event extraction to be applied
to an open range of text. However, due to the free-form nature of creative writing,
events will usually not follow a set grammatical or semantic rule or structure that can be detected and extracted, making such methods a poor match for creative fiction text.

One of the earliest attempts at event or information extraction applied to the fiction domain is a paper written by Sharon Givon of the University of Edinburgh where central characters and their relationships are extracted (Givon 2006). A lot of research has followed in the extraction of characters relationships and social network interactions in fiction (Elson, Dames, and McKeown 2010; Agarwal, Kotalwar, and Rambow 2013; Dekker, Kuhn, and van Erp 2018), and even ontology creation (Goh et al. 2011), but little research exists pertaining to the automated extraction of fiction plot. Hajime Murai has researched the behavioral and emotional aspects of characters in microfiction in an attempt to use their vocabulary and behavior to model the plot structure with the goal of developing an automated plot extractor (Murai 2014, 2017). Later he attempts to extract plot from detective comics (Murai 2020). This method of plot extraction is on the thematic level similar to how Vladimir Prop viewed plot, and its purpose is to find transition patterns from one thematic element to another, displaying them in a relational graph structure similar to the research done in social network extraction.

3. Narrative as a Scatter-plot of Entities

Visualizing a story as a plotting of points or lines is not a new concept. Many such graphs are created through sentiment analysis (Nalisnick and Baird 2013; Jacobs 2019) to give a visual of the narrative as it progresses through the story. Here I solely use the entities of the story to visualize the story content in a scatter-plot and then cluster those entities in an attempt to isolate the individual scenes of the narrative.

3.1 System Outline and Explanation

Figure 1 shows the diagram of the Scatter-plot of Entities. The input to the system is a plain-text document of the story. It can perform without modifications to the document, but for better accuracy, hand-coreferenced stories are used to ensure that any mention of an entity as a pronoun or other moniker will be recognized as that same entity within the text. An NLP pipeline then handles the preprocessing of the text by tokenizing it, tagging parts of speech, and then running named entity recognition (NER). NER identifies names of people, places, and organizations but also dates and other text with

![Figure 1](image-url)
specific formats. For the purposes of this research, only those entity types that can correlate to actors, object, and locations in a text that could potentially be plot-important are chosen from the list generated by NER. Appended to this list are common objects and pronouns that are missed by NER, such as the first-person pronoun ‘I’, which is important in first-person narratives where the speaker’s name is not given and thus cannot be coreferenced.

Once the list of entities is created, they are located within the text and given an x-coordinate number corresponding to their narrative-order location. A y-coordinate is provided according to the order of that entity’s first appearance within the text. Each instance of a same entity will have the same y-coordinate but a different x-coordinate depending on where within the text that specific instance appears. Those entities that only appear once or twice within the entirety of the text are removed from the list both to reduce clutter and due to the assumption that if it appears that little, it is not as important to the narrative. Using these (x,y) coordinates, the entities can be plotted on a Cartesian plane.

A density line is then created with respect to entity x-coordinates. This is done by creating a Gaussian curve for each entity centered at the x-coordinate of that entity, creating a distribution overlap of entity positions that are then summed together to create a density curve. The more entities that appear near each other in the text, the more overlap is present and the higher the value of the curve at that point. The y-values of this curve serve as the density values and can be seen as a heat-map of narrative intensity, meaning many entities are involved.

Clustering is used to segment the text into scenes based around entity appearance. A single-dimensional Mean Shift algorithm variant is used for this purpose. Mean Shift uses the density gradient of data coordinates to attract data points together, climbing the gradient slope to where the points are densest. A bandwidth value sets the distance of attraction for the points, determining the number of clusters created. The instances of data for the clustering algorithm are the instances of the entities within the text, and the feature by which these entities are clustered is their x-coordinate, their location in the story. The reasoning behind this is that the more entities that are near each other in the text, the more these entities are interacting within the story. The naive assumption directing this method is that important entities interacting with each other hint at an important scene happening in the story at that location, and the borders of those events where the intensity is low represent the divisions between scenes.

Clustering is used instead of calculating local maxima of the density line because due to the nature of the curve’s creation, there are often many local maxima and minima, far more than there are scenes in the story. Using a Mean Shift clustering method, the entities converge into clusters centered around the largest grouping of nearby maxima. Instead, the density line is used to tune the clustering. The borders between the clusters are calculated at half the distance between the nearest two entities in neighboring clusters. This creates a perfect split between clusters because they are clustered unidimensionally. The x-coordinates of those borders are then input into the density line, and the resulting y-value, the intensity, is retrieved. The bandwidth of the Mean Shift algorithm is iteratively adjusted searching for the clustering that places the borders between clusters at the lowest intensities averaged over all borders, making the center of the clusters where the intensity is highest. This way, every location of local high intensity is in its own cluster, creating a partitioning of the narrative around these locations of high intensity, which become the scenes of the story.
Scatter-plot of Entities for the Short Story “Falling”

Figure 2
Entity scatter-plot (top) and corresponding heat-map (bottom) for the short story Falling (DeBuse 2013). Story entities are plotted as black dots, where the x-value is the location in the story, and the y-value is the order of the entity’s first appearance. Vertical red lines show the cluster partitions. Vertical blue lines show the actual scene transitions.

3.2 Results

Figure 2 shows the output of the Scatter-plot of Entities system for the short story Falling (DeBuse 2013). Two visuals are included in the output, the entity scatter-plot on top and the heat-map on bottom. On the scatter-plot, the x-axis is the narrative order of tokens in the story, so it represents the location of each entity in the text. Each tick on the y-axis represents a unique entity in the order that entity is first introduced in the text, meaning the very first entity would have a y-value of 0, where the last would have a y-value equalling the number of unique entities found by the system. The arrangement of the y-axis in this manner is solely for visual and informative effect so that all entities are not all plotted on the same unidimensional number line. This creates interesting visual artifacts in the scatter-plot. First is the ability to see at a glance where certain main entities are involved in scenes in a story and where they are not. The main character of Falling is entity 0, and the secondary main character is entity 10. Another prominent character is entity 51. The involvement of these characters together in various scenes is noticeable, and the vertical red lines delineating the partitions of the clusters do well marking these locations. Second is how the introduction of entities follows almost a logarithmic curve. Figures 3 and 4 show a comparisons of the output for William Shakespeare’s Hamlet script (Shakespeare, Raffel, and Bloom 2003) and Disney’s The Lion King script (Allers and Minkoff 1994). In general we can see a common narrative pattern, that after the first
Figure 3
Scatter-plot of Entities for the script of Hamlet. The number of entities the system detects is far greater than The Lion King, as it is a more complex story, making the scatter plot more dense. However, the curve of the introduction of the entities is very similar, finishing the introduction of most of the entities in the first quarter for both stories.

Figure 4
Scatter-plot of Entities for the script of The Lion King. The Lion King begins with a lot of description and a song before the introduction of the first main characters of the story. This is why the most prevalent entities that create the densest horizontal plots are higher than in Hamlet where the characters are introduced right at the start. The introduction of Timon and Pumba is also immediately visible near the top as almost parallel plots for those two entities because they are often referenced as a pair, “Timone and Pumba,” in the script.
quarter to third of a story, the introduction of new entities slows considerably. Different arcs of the story are also visible solely by noting the presence of prominent entities, seen as heavy, horizontal black-dotted lines.

Like the scatter-plot, the heat-map’s x-axis is the location within the text. The y-axis is the intensity of entity density determined by the summation of overlapping Gaussian curves. Included on the bottom is a colored visual of the heat map where blue is low intensity and yellow is high intensity. What is interesting to note in Figure 2 is that the two locations of greatest intensity in the heat-map (in the middle and at the end) are the two scenes of greatest importance within this particular story. However, this is not always the case for every story.

The red lines running vertically through each plot are the partitions of the clusters, where the system believes the scene transitions are located. The blue lines are the ground truth borders between the scenes. We can see in Figure 2 that for the first five scenes, the system nearly matches the ground truth, but for all borders in the last five scenes save one, the borders missed completely. The system even combined the climax and resolution scenes into a single scene, missing the last scene transition at the end. This is likely due to the clustering method. The sixth scene is so long within the text that the bandwidth is not large enough to fully attract all the entities within its breadth forcing an early division that offsets the scene divisions later on; increasing the bandwidth to compensate for this combines some of the shorter, earlier scenes together.

3.3 Discussion

Although the visual artifacts of the Scatter-plot of Entities are intriguing, extensive testing such as the statistical analysis of many different stories was not performed due to the system falling short of the desired goals of plot and narrative flow visualization. The characters within the story are shown as entities, and events as scenes are isolated, but those are the only two aspects of plot that this system accomplishes. There is no way to determine the links between the scenes and why they follow the scenes that came before. By creating a partition as such, all that is created is a listing of scenes. No causation is present. Without that linking, a true plot structure cannot develop.

This does not mean the Scatter-plot of Entities has no merit. There are narrative patterns that can be investigated further in the scatter-plot format. One direction for future research involves delving further into this type of narrative visualization to see what forms different media produce, such as news articles, academic text, and others and whether or not the type of medium can be identified. Comparisons between stories could also be performed to detect similarities in narrative through the involvement of entities throughout the stories.

4. Narrative as a Graph

To better visualize narrative flow in a manner that could emphasize the plot of the narrative, the causal element of plot needs to be addressed. As mentioned earlier, plot is more than simply the entities of the story and the events in which they are involved. Linking these events together to show how the narrative flows would better represent plot. Treating events as vertices and their connection through the narrative as edges, a graph structure can be implemented as a mapping of the narrative.
4.1 Narrative Flow Graph Definition

A narrative flow graph $M$ is defined as $M = (S, E)$ where $S$ is a list of scenes in narrative order and $E$ is a set of all edges of the graph. Let $\phi$ be an entity within the text and $\Phi$ be the set of plot-important entities as defined by the user glossary. Then $s \in 2^\Phi$ where $s$ is a scene in the story and $2^\Phi$ is the power set of $\Phi$. An edge $e$ is defined as $e = (s_a, s_b, \phi_i)$ where $a$ and $b$ are indices into the ordered list of scenes, and $\phi_i$ is a specific entity attributed to that edge.

The restrictions involved in edge creation are what make the graph structure a narrative flow graph. The five restrictions are as follows:

1. $(s_a, s_b) \in S^2$
2. $a < b$
3. $\phi_i \in \Phi$
4. $\phi_i \in s_a \cap s_b$
5. $\nexists x | \phi_i \in s_x, a < x < b$

Restriction 1 simply states that both scenes $s_a$ and $s_b$ are scenes in the story. The second restriction forces the graph to be a directed graph moving forward in narrative order, and no vertex can have an edge to itself. The third requires the entity $\phi_i$ to be plot-important as defined by the glossary. The fourth and fifth restrictions are the most important in defining edge creation to create the causal connections between scenes in the story. In 4, an edge is created if the two scenes $s_a$ and $s_b$ share the same entity $\phi_i$. An edge can only be defined by a single entity, so an edge is created for each entity involved in both scenes. Lastly in 5, scene $s_a$ must be the most recent occurrence of $\phi_i$ previous to scene $s_b$. The true meaning of “causation” when applied to the narrative flow of scenes in a story and whether or not restrictions 4 or 5 sufficiently encompass this meaning can be argued; however, there is certainly a connection between proceeding scenes in a story involving the same entities, and that connection alone is sufficient for the purposes of this research in defining causation. This creates a directed acyclic graph (DAG) showing scene dependencies, what scenes must occur before a following scene as determined by the entities involved in those scenes. Additionally, due to the narrative ordering of scenes, the adjacency matrix for the narrative flow graph is naturally a strictly upper-triangular matrix without the need for permutation.

4.2 Annotation and Data Preparation

The Plot Map System attempts to automatically generate a narrative flow graph given a source story, but to do so the story source text first requires some data preparation before it can be used as input. Before detailing the system, the preparation methods and the reasons for them should first be discussed.

4.2.1 Entity Glossary. In order for the resulting narrative flow graph to be clean and understandable, all unnecessary entities need to be ignored, focusing only on those actors, locations, and objects within the narrative that are plot-important. Determining plot-importance computationally is a monumental task that relies on more than NER and entity frequency as was used in the first method in section 3. In simple terms, NER is insufficient.
Off-the-shelf NER tools have made great improvements over recent years, but a large margin of error remains, especially when applied to fiction text. In a study by Dekker et al. on NER for use in social network extraction in fiction text, they tested four of the top available NER systems and found that there remains a large variance in accuracy in NER between novels (both classic and modern) with the precision of some being as low as single-digit while others are near-perfect (Dekker, Kuhn, and van Erp 2019). In addition, they identified a few of major issues that fiction stories pose to NER, such as novels in 1st person performing significantly worse than novels written in 3rd person, difficulty in recognizing names with non-alphabetic characters such as apostrophes, and word names (names that are the same as common words) being unrecognized about 50% of the time. When considering plot-important entities, common objects and locations can also be important to a story’s plot (such as a kitchen if the scene takes place in a kitchen or a knife if that knife is the murder weapon in a mystery novel). Similar to word names, these will not be detected by NER systems. Cuong Xuan Chu et al. created the ANTYFI system to perform entity typing on fiction text (Chu, Razniewski, and Weikum 2020) which is the first step in competent NER for fiction domains, but more research is needed in this area. Lastly and most importantly, NER is incapable of assessing plot-importance, and the field of research in the automation of determining plot-importance of entities in a story is underdeveloped.

Due to the difficulty of the task, automatic detection of plot-important entities was deemed beyond the scope of this current research. Using existing tools for entity detection would only serve to add a large amount of error into the output. Instead, the user of the system creates a glossary of entities they would like the system to follow. Generally, all entities the user decides are plot-important are included in the glossary; however, should the user choose to do so, certain entities may be omitted to force the narrative flow graph to follow only those entities the user desires, streamlining or tuning the graph’s focus to the user’s needs.

The glossary is coded in XML using the format shown below:

```xml
<glossary>
  <entry id="#" type="???">
    <label>moniker</label>
  </entry>
</glossary>
```

Each entry in the glossary includes an ID number, type, and list of labels.

- **id**: unique number assigned to this specific entity.
- **type**: either “actor”, “location”, “object”, or "other"
  - actor: entities of volition within the narrative
  - location: physical locations where events take place
  - object: entities without volition
  - other: entities that do not fall into the above three categories
- **label**: words by which the entity is uniquely referred in the text. The first entry is used on the plot map as the label for corresponding edges as well as for replacement within the text when coreferencing.
4.2.2 Coreferencing. Similar to the stories used in section 3, the stories used for input into the Plot Map System are coreferenced by hand. There are two reasons. The first is due to the inaccuracy of available coreferencing tools, the top of the line for both machine/deep learning and rule-based methods being little better than 80% in $F^3$ precision but mostly in the 60’s and 70’s (Sukthanker et al. 2020). Second, the scope for coreference detection is generally small, meaning that even though a text may be long, available tools still only use local passages within the text to determine the reference. Existing coreference resolution tools are incapable of knowing that one reference early on in a long story is the same as a reference near the end of the story, especially if that reference is mentioned by a different word or name. For this system to be accurate and properly locate the desired entities in the story, it must know every instance of an entity from the user glossary no matter where, how, or in what manner it is mentioned in the source text.

Coreferencing is done in CSV files where the first column contains every token of the story in narrative order, and the second column is for the ID number from the glossary to which that token is referring if applicable. For example, the "he" token on one sentence might not refer to the same entity as the "he" token in another. Similarly, if there are multiple entities referred to by the same words, such as if there are two people by the same name, each entry in the 2nd column would contain the corresponding ID number from the glossary for each separate entity. This way each reference in the text to an entity in the glossary is correctly matched to that entity. Those tokens are then replaced with the first label of that entity’s entrance in the glossary.

4.2.3 Location Ground Truth. In addition to data preparation, ground truth data is created to provide comparison with the system’s output. The first is scene location. As part of the system’s process detailed below in section 4.3, inference of the locations where events take place is performed. To ascertain the accuracy of this inference, a CSV file is created similar in form to what is used for coreferencing. In the first column are the tokens of the story, and in the second column are the ID numbers from the glossary associated with the locations at which the events occur. Being similar in form to the coreference annotation, the annotators completed both the coreference and location annotations simultaneously for each story.

4.2.4 Narrative Flow Graph Ground Truth. Lastly, the output of the system needs a ground truth comparison to determine its accuracy. The ground truth narrative flow graphs are created according to a human’s understanding of the plot, only including those scene that are plot-relevant and excluding all others as well as dividing the scenes where they believe the scene partitions exist. This is done by creating an XML file with all the necessary information needed to generate a narrative flow graph in the same form as what the system would produce.

The glossary is coded in XML using the format shown below:

```
<map>
  <nodes>
    <node id="#" location="id#" summary="this is the first scene description" sentences="1-26">
      <entity>entity id</entity>
      <entity>entity id</entity>
      <entity>entity id</entity>
    </node>
  </nodes>
</map>
```
The XML file is separated into two groups: the nodes and edges required to create the graph structure. Included in each is the pertinent information on what is stored in those data types and how they connect.

• **node**
  - id: unique integer to differentiate between entrances in the glossary
  - location: the id from the glossary of the location the scene represented by the node takes place
  - summary: a brief summary of the scene represented by the node
  - sentences: the sentences of the text, numbered by narrative order, that the scene takes place
  - entity: nested within node, it is a listing of the IDs from the glossary of the entities present or involved in the scene represented by the node

• **edge**
  - id: unique integer to differentiate between entrances in the glossary
  - entity: the ID from the glossary of the entity the edge represents
  - description: a brief description of the causal relation of the edge linking the scenes
  - previous: the ID number from this same XML document of the node the edge leaves
  - next: the ID number from this same XML document of the node the edge connects

The XML file is read by the system and used to produce a narrative flow graph in the same form as if the system had extracted this information from the text so that the comparison between the system’s output and the ground truth annotation is as clean as possible.

### 4.3 System Outline

The purpose of the Plot Map System is to generate a graph structure following the definitions and restrictions in section 4.1. Figure 5 shows the diagram of the system. It is divided into three main sections each with its own intermediary output. Each process of the system is explained in detail below.

#### 4.3.1 Data Preprocessing

Data preprocessing takes as input the plain text file of the story, the glossary XML, and the coreference CSV. The glossary is used to generate a list of entity objects wherein is stored the entity’s unique ID, the entity’s type, and all labels to which the entity is referred in the text. The raw text is then tokenized to isolate the
instances of the entity references in the text. Using the entity list and coreference CSV, each instance of an entity is replaced with the first label attributed to that entity. The coreferenced story is then outputted to be used in the next step of the system.

4.3.2 Early Graph Generation. Early graph generation takes as input the coreferenced story and the entity list. Following a simple algorithm, an ordered list of vertices and a set of edges are created.

**Step 1:** Parse the text into sentences and detect which sentences involve entities from the list. These sentences become the events of the story the initial graph is built around.

**Step 2:** Create a vertex for each event storing the sentence number, text, and involved entities. These vertices are added to an ordered list by their appearance order within the text.

**Step 3:** Following the definitions and restrictions in section 4.1, create edges between each vertex by iterating through the list of vertices, and for each vertex search backwards through the list of vertices for the most recent appearance of every entity within that vertex. Each entity creates an edge to the most recent vertex where the entity last appears unless the current vertex is that entity's first appearance.

4.3.3 Narrative Flow Graph Creation. The graph at this stage is extremely large and difficult to reason about for long passages of text because the graph is comprised of event vertices made from individual sentences. Although a graph created with such high fidelity to the source text can be informative, to better notice patterns and features of the narrative, the graph needs to focus on the relationships between scenes. Individual sentences alone do not represent scenes in a story. Scenes in narrative can be described as an unbroken chain of events involving similar entities within a the story. The definition of a scene in section 4.1 is sufficient for the narrative flow graph’s structure, but to refine a graph structure of event vertices into a narrative flow graph made of scene vertices, further definitions are needed.

An entity $\phi$ is defined as $\phi = (i, \tau)$ where $i$ is the ID of the entity according to the user glossary and $\tau$ is the entity type. For an event $\varepsilon$ we can state $\varepsilon \in 2^\Phi$, because like scene $s$ it is comprised of entities. $\mathcal{E}$ is a list of events in narrative order. An event graph

---

Figure 5
Diagram of the Plot Map System.
$M^*$ is defined as $M^* = (E, E)$. We can now define a function

$$f : M^* \rightarrow M$$

performing the following processes:

1. Until $E = \emptyset$, create consecutive lists $E'_{n-m}$ from $E$ where $E'_{n-m} = \langle \varepsilon_n \text{ to } \varepsilon_m \mid \forall \varepsilon \exists \phi$ where $\tau \in \phi$ is of type: "location" and $\phi_n = \phi_{n+1}$ until $n = m \rangle$. The original indices of all $\varepsilon$ are remembered.
2. Create a scene $s$ for each consecutive list where $s = \{ \phi \mid \phi \in E' \}$. $S$ is the set of all $s$.
3. For all $e \in E$ and $e = (\varepsilon_a, \varepsilon_b, \phi)$, if $a = i$ where $i$ is the original index of $\varepsilon \in E$, set $a = j$ where $j$ is the index of $s \in S$.
4. For all $e \in E$ and $e = (\varepsilon_a, \varepsilon_b, \phi)$, if $b = i$ where $i$ is the original index of $\varepsilon \in E$, set $b = j$ where $j$ is the index of $s \in S$.
5. Remove all $e$ from $E$ where $s_a = s_b$.

The purpose of function $f$ is to take the single-sentence events and group them together creating scenes by combining those consecutive events that take place in the same location. All entities involved in those events are added to the new scene, and any edges attached to those events are changed to attach to that scene. The main limiting factor of this function is that not all sentences have an explicitly stated location, therefore not all events will have a location entity within it. This complicates step 1 of the function. To overcome this, location inference is performed before scene creation.

Location inference is the process of determining the location of events using the last known location searching backwards through the edges of the graph. A breadth-first search is performed in reverse edge direction for each event vertex without a location entity. Once a location entity is discovered, the entity is applied to the source vertex that began the search. This is better than simply finding the most recent location stated in the text because those branches of the narrative that have no relation to the current event (there are no entities in common) will not affect the location inference of that event. This is beneficial for narrative structures where separate event or scene branches are running in parallel.

There remains a single issue with this method of scene creation, and that is when multiple scenes happen at the same location. By strict adherence to the mathematical definition, all those scenes are combined into one. Essentially, the definition assumes that there is a single scene per location. While that may be the case for some stories, it certainly does not hold for all. To address this in the output, the scenes are not fully combined. Instead, the overall scene is separated into pieces where entities differ between parts of the scene. Only those consecutive events where entities are similar are combined. While this does increase the complexity of the narrative flow graph—making more vertices and edges—it also makes the graph more informative, essentially finding a middle ground between the $M^*$ and $M$ depending on how correlated the entities are within the scenes. If the actions and interactions between entities in the scenes are dissociated, the output graph will be closer to $M^*$. Where they are more unified, the output graph will be closer to $M$. In the results, you will see the difference between the output where scenes are more divided and the ground truth narrative flow graphs.
where all events at a single location are combined into one scene, particularly in *The Sound of Thunder* where there are only three scenes. One negative side effect of this further scene breakdown is that dialogue creates a large number of vertices, often a vertex for a single line someone speaks. For scripts which are almost solely dialogue, the number of vertices is far larger than short stories even when taking the length of the text into account.

Once the scenes have been created and edges affixed to the correct scene vertices, text summarization is performed to create a more informative plot map. This is done through a TF-IDF algorithm. A matrix of comparisons between each sentence in a scene is created. Those sentences that score the best overall for comparison with all other sentences in that scene are most likely to represent the scene as a whole. This is a simplistic approach, and a neural text summarizer would perform better, but that is a topic for future work.

Finally the graph structure and pertinent information such as edge and vertex labels and scene summaries are formatted into a ".gv" file to be compiled and interpreted by GraphViz’s dot compiler (Ellson et al. 2003). Edges and vertices are also given color according to the entities each edge represents or the location the scene takes place. The narrative flow graph is formatted in a left-to-right orientation with each consecutive scene further right than its predecessor. This prevents any doubling back that would complicate the visual.

### 4.4 Results

Seven stories were selected and run through the Plot Map System.

3 professional short stories: *Leinigen Versus the Ants* (Stephenson 1972), *The Sound of Thunder* (Bradbury 2016), and *To Build a Fire* (London 2007)

2 amateur short stories: *Observer 1: A Warm Home* (DeBuse 2012a) and *Observer 4: Legends* (DeBuse 2012b)

2 scripts: *Hamlet* (Shakespeare, Raffel, and Bloom 2003) and *The Lion King* (Allers and Minkoff 1994)

The resulting narrative flow graphs are quite wide and detailed images that cannot fully be shown in a typical paper. As such, sections of various output graphs are provided as examples of common artifacts seen in most narrative flow graphs as well as to show comparisons between the graphs. In the example images below, the most important areas of focus are the vertices and edges. Due to the restrictions of page size, some of the text will be too small to see well for those images covering a large section of the graph.

#### 4.4.1 Common Structures within Narrative Flow Graphs

Each narrative flow graph is rather distinct in its overall shape and appearance, but within are common structures that are repeated over and over again in many different narrative flow graphs, giving hints to common, recurring narrative patterns. I have selected names for each based on their shape and function: Braids, Parallels, Breaks, Convergent Points, and Ends. Braids and Parallels deal with passages of narrative, Breaks and Convergent Points deal with scene transitions, and Ends deal with the end of a narrative branch.

Figure 6 shows examples of different braids from different stories. Braids, like the name suggests, involve the interweaving of specific entities important to a scene or consecutive scenes as all other entities pass by to where they come into play later in the story. These show areas where the entities involved in the current events of the narrative
Examples of Braid structures in output graphs for six different stories. Braids are the most common of all structures in a narrative flow graph, created by the interaction of a finite number of entities involved in consecutive events in scenes. Braids can include a small number of entities as in Simba and Nala’s interaction in *The Lion King*, or many entities as seen with the hunters in *The Sound of Thunder*.

Figure 7 shows an example of a Parallel. Parallels involve alternating scenes where the entities involved in one scene are not involved in another, creating almost a tiered structure within the narrative flow graph. Where parallels are prevalent, after one part of the parallel is finished, the next scene often involves entities of the scenes that came before, as seen in the Lion King script where the imprisoned Zazu sings to Scar, and the hyenas report that the lionesses will not hunt. This happens in parallel with the scenes of Simba in the jungle with Timon and Pumba. Breaks are the most common scene transitions for parallel scenes, as a line can almost be drawn between them. For breaks, there is little to no interaction between entities of the two scenes, so the transition is a breaking of the narrative of one scene giving way to the start or continuation of the narrative of the following scene.
A perfect example of parallel scenes in *The Lion King*. On the left and right in green are the jungle scenes with Timon, Pumba, and Simba. In the center in yellow is the scene where the imprisoned Zazu sings to Scar. There is no overlap between entities in these scenes, and the locations are different, creating clean scene Breaks. This is beautifully reflected in the narrative flow graph.

Convergent points, unlike breaks, are where many entities converge to begin the next scene (high local edge density). As stated before, the involvement of many entities does not necessarily mean that scene is more plot-important; however, many crucial scenes in the story often do involve multiple entities. The largest example of this is climaxes. In the narrative flow graphs generated by the system, many climaxes are visible at a glance solely by looking for convergence in entities (many edges leading to the same scene or between groups scenes) near the end of the story. In addition, crisis points (intermediary sub-climaxes) are visible the same way. Figure 8 shows examples of convergence at climaxes. In the captions for each figure, an explanation of the convergence for that story is given. The local edge density changes of course depending on the story.

Interesting to note here is that by looking at the outgoing edge density of the vertices, the buildup of the story to the climax and winding down at the resolution is visible. Figure 9 shows the outgoing edge density of the ground truth narrative flow graph for *The Lion King* where the climax is the highest spike in outgoing edge density: the fight for Pride Rock. There are some intermediary crisis point midway through the story representing the Elephant Graveyard scene at vertex 9 and the wildebeest incident leading to Scar telling those back at pride rock that both Simba and Mufasa died in vertex 13. Incoming edge density can give similar information, but the benefit of recording outgoing edge density is that the ends are visible as values of zero (no outgoing edges).

Ends are exactly what they imply, the end to a branch of the narrative. These are sinks in the narrative flow graph. The most common end is the ending of the story, but not all ends have to be the ending of the story. Figure 10 shows the death of Scar in *The Lion King* script as well as the end of the story itself. The entities involved in those events, namely Scar and the hyenas, converge on that end and go no further. Their involvement in the story ends there.

**4.4.2 Ground Truth Comparisons.** Ground truth narrative flow graphs are created following section 4.2 with the goal of demarcating the actual scenes of their respective stories to show the relationship between events and the entities that connect them according to human understanding of the plot. Figure 11 shows the entire narrative flow graph for *The Lion King* as an example of the ground truth graphs. They are
Top left: Climax of *Observer 4: Legends* where the spirits of the fallen warriors witness Evelyn becoming legend. Top right: Climax of *Hamlet* where Hamlet, Laertes, and most of the cast have died. Bottom: Climax of *Observer 1: A Warm Home* where the village mob burns the cottage down and pursues the orphan.

---

The outgoing edge density of the ground truth narrative flow graph of *The Lion King* showing the steady rising intensity of the story until the climax, and then ending with the resolution. The shape beautifully reflects a Fichtean Curve (Gardner 1991). The crisis point at vertex 9 is the Elephant Graveyard scene. The crisis point at vertex 13 is the announcement that Simba and Mufasa were killed by a wildebeest stampede. The climax at vertex 27 is the battle for Pride Rock.
Figure 10
Distant view of the end of *The Lion King* in the output narrative flow graph showing two end points (highlighted with red boxes). The first near the center is the death of Scar. All entities involved in this, namely Scar and the hyenas, have no part in the narrative beyond this point, creating an sink. The second red box on the right is the end of the narrative flow graph, a sink where all other preceding vertices and edges eventually lead.

Figure 11
Distant view of the entire ground truth narrative flow graph for *The Lion King* to show vertex and edge relationships.

Figure 12
Distant view of the latter half of the ground truth narrative flow graph for *The Lion King*. The section highlighted in red represents the same parallel shown in Figure 7. The section highlighted in blue is the climax where all entities within the story are involved in the battle for Pride Rock, another example of the convergent points shown in Figure 8. The section highlighted in green shows the same two end points shown in Figure 10. The shape is also nearly identical except for the inclusion of a few more vertices in the output plot map: 2 leading to Scar’s death end point, and 5 leading to the end of the story.

less complex, and thus shorter, than the graphs output by the system due all events being fully combined into their respective scenes. This can be seen in Figure 12 where the parallel scenes from the output in Figure 7 are shown in the red box as single vertices and the end points in Figure 10 are shown in the green box. The shapes of both these structures between the output and ground truth narrative flow graphs of *The Lion King* are remarkably similar, showing for this story the system does well correctly representing the plot and narrative structure at those locations.

In the results below, the comparisons for each story are both visual and numerical with analysis performed on how many scenes in the ground truth graphs are repre-
Ground Truth Comparisons and Accuracy

<table>
<thead>
<tr>
<th>Story Title</th>
<th>Length (words)</th>
<th>Ground Truth Scene Count</th>
<th>Output Vertex Count</th>
<th>Location Acc</th>
<th>Scene Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Sound of Thunder</td>
<td>4364</td>
<td>3</td>
<td>141</td>
<td>0.425</td>
<td>0.397</td>
</tr>
<tr>
<td>To Build a Fire</td>
<td>7104</td>
<td>10</td>
<td>74</td>
<td>0.527</td>
<td>0.228</td>
</tr>
<tr>
<td>To Build a Fire*</td>
<td>7104</td>
<td>10</td>
<td>74</td>
<td>0.926</td>
<td>0.68</td>
</tr>
<tr>
<td>Leinigen Versus the Ants</td>
<td>8666</td>
<td>18</td>
<td>156</td>
<td>0.801</td>
<td>0.551</td>
</tr>
<tr>
<td>Observer 1: A Warm Home</td>
<td>7254</td>
<td>35</td>
<td>185</td>
<td>0.555</td>
<td>0.391</td>
</tr>
<tr>
<td>Observer 4: Legends</td>
<td>11130</td>
<td>35</td>
<td>254</td>
<td>0.248</td>
<td>0.257</td>
</tr>
<tr>
<td>Hamlet</td>
<td>32065</td>
<td>35</td>
<td>642</td>
<td>0.837</td>
<td>0.553</td>
</tr>
<tr>
<td>The Lion King</td>
<td>15765</td>
<td>31</td>
<td>567</td>
<td>0.545</td>
<td>0.546</td>
</tr>
</tbody>
</table>

Table 1: Vertex counts and accuracy for the seven stories run through the Plot Map System. The second To Build a Fire calculation (denoted with *) is using simplified location annotation.

sent in the graphs outputted by the system as well as how well location inference of scenes is performed by the system. Scene accuracy is determined by calculating the overlap of scenes from the output to the scenes in the ground truth based on location in the story, not vertex count, and whether the information stored in those output scenes match that of the ground truth. Scene summary is excluded from this calculation. Location accuracy is determined by how well the location inference matches the location ground truth, calculated by simple matching percentage. Table 1 shows the resulting accuracy calculations as well as story length in words, the number of scenes in the ground truth narrative flow graphs, and the number of vertices in the output graphs.

4.5 Discussion of Results

The Sound of Thunder: This story is unique among those chosen in that it only contains 3 scenes in the entirety of the short story: the present before traveling back in time, the time in the past where the hunt takes place, and the return to the present (see Figure 13). This created a unique challenge for the system. The number of vertices in the output is the 2nd smallest, but it is still many times greater than the number of scenes. In addition to this, a number of the plot-important objects to show the change from the original present to the changed present at the end of the story are simply mentioned in the text without any direct connection to actors in the scenes or references to the scenes location. Because of this, they appear in the plot map without any scene or location labels until a connection is made. This drastically reduces the accuracy of those scenes where these objects first appear. And with only 3 scenes for the story, the overall accuracy takes a heavy hit. For location accuracy, the locations chosen by the annotators were the “future/present” period and the “ancient wilderness”. Having only two locations should make inference rather simple if not for the fact that both locations are discussed throughout the story when the characters are both in the present and ancient past. The Plot Map System does not have the capability to differentiate between when the actors are physically present in the location and when a location is discussed in conversation or described in the text. As such, when these locations appear in the text, the system mistakenly thinks in a number of the vertices that that is where this section of the story takes place, decreasing the location accuracy for those vertices. This story was selected due to its overall simplicity, but even with such a simple story, perfectly representing it in a narrative flow graph proved to be a difficult task.

To Build a Fire: This story was chosen due to its difficulty in location assessment. Much of the story takes place along Henderson Creek, but the events occur at specific locations
along and off that creek’s trail. The system correctly assumes that most of the story takes place along Henderson Creek, but it is often incapable of determining the specific locations the ground truth plot map denotes as the location many of the scenes take place. All of these complicate correctly constructing a narrative flow graph where location is paramount for scene creation and description. This is the main reason scene accuracy is the lowest. If the need to know the specific locations along Henderson Creek is removed (as seen in Table 1 with To Build a Fire*), the location accuracy increases to 0.926, and the scene accuracy increases to 0.68, making it the second highest location and scene accuracy.

Leinigen Versus the Ants: This short story was chosen as a perfect example of buildup within a story, meaning that as the story progresses, the number of entities involved and the frequency of that involvement increases. This can be seen in Figure 14. The braids in the earlier parts of the story involve fewer entities, but as the story progresses (the assault of the ants becomes more dire), more and more entities become involved, both actors and objects. The climax at the end is locally very dense in the number of edges connecting the last few nodes before the thinning down at the resolution (another beautiful example of a Fichtean Curve as seen in Figure 15). The Plot Map System does well for the majority of the story matching both scene and location until the end where it has difficulty ascertaining where the final scenes in both the climax and resolution take place. The majority of the location accuracy comes from the first 2/3 of the output plot map. Similarly, due to the complexity of the climax, the system struggles to accurately partitioning the scenes.

Observer 1 & 4: The Observer short stories were chosen because they are amateur works, to see how the Plot Map System handles a less-professional or less-refined narrative. Observer 1: A Warm Home is simple in its locations, all taking place at a single cottage on a hill. Like with To Build a Fire, there are specific locations within locations

Figure 13
The ground truth narrative flow graph for The Sound of Thunder. The story only has three scenes, two in the present/future and one in the ancient past. Because of this, the ground truth plot map is less informative, only showing what entities within the story are involved in the past or present and showing little about their interactions.
Figure 14
The ground truth narrative flow graph for *Leinigen Versus the Ants*. The braids in the earlier parts of the story involve fewer entities, but as the story progresses the density of the braids and complexity of their interactions increases. This is a perfect example of buildup in a story.

Figure 15
The outgoing edge density of the ground truth narrative flow graph of *Leinigen Versus the Ants*. The crisis point at scene 5 is when the ants reach the plantation and countermeasures begin. Scenes 9 and 10 are when the dam fails to keep the ants back, and everyone retreats. The climax at scene 13 where local density is highest is where Leinigen has no other choice but to flood his plantation.

on the hill and in the cottage where events happen, but they are more explicitly stated, making it easier for the system to detect. The difficulty in this story is that not every event is plot-important. There are scenes that give details about characters or locations or scenes with interactions between characters that do not contribute to the buildup and conclusion of the story. Because of this, there are a number of vertices in the output that do not have a corresponding scene in the ground truth plot map, decreasing the accuracy. *Observer 4: Legends* is a hero’s journey story with a large climactic battle at the end. Locations are varied and change as the story progresses, never returning to a former location until a time-skip in the resolution. This is visible in the resulting narrative flow graph as the color and location labels on the vertices change as the graph progresses from left to right. The system has difficulty getting every single location correct, especially because they are often not explicitly stated as the characters travel. In addition, in the climactic battle at the end, events happen at different locations on the battlefield, and similar to *To Build a Fire*, the system has difficulty correctly inferring those specific locations. Combining those two challenges, the location accuracy is the worst of all the stories. Scene accuracy is also poor because there is a lot of exposition.
talking about the past and distant locations. While these parts are plot-important for
the story, the system has difficulty knowing when a flashback or exposition ends and
continues those into proceeding scenes until something within the text signifies that
a new scene has started instead of the continuation of the scene before the flashback
or exposition. This muddies scene boundaries, making it difficult to tell where many
scenes in the output begin or end, drastically decreasing the scene accuracy.

**Hamlet Script:** Play scripts differ greatly from short stories. Very little information is
given about place and setting, and actions are either very simply described (such as a
character leaving or entering) or otherwise are not mentioned at all, being left to the
performers to determine what actions best fit the scenes. The main bulk of the text are
lines spoken by the characters, and each line explicitly states who speaks it. This makes
determining what entities are involved in a scene a very simple task. Likewise, because
the script is broken directly into scenes, each scene being its own specific location in
the script (where the actual location where events take place in the story is often not
mentioned at all), creating scenes that match the ground truth is also very simple.
Because of this, the narrative flow graph outputted for the *Hamlet* script had the highest
accuracy for both location and scene when compared to its ground truth.

**The Lion King Script:** This cinema script differs from the play script of *Hamlet* in that in
addition to lines spoken by the characters, the scenes, locations, and actions taken by the
actors are explained simply. This creates almost a middle ground between a play script
and one of the other short stories. Figures 7 and 10 above show parts of the output
for *The Lion King* script, while Figure 11 above shows a distant view of the ground
truth narrative flow graph in its entirety. One challenge of this script is the musical
numbers. While some of the information in different numbers is plot-important, their
main purpose is entertainment. As such, not all of the musical numbers are included
in the ground truth narrative flow graph; however, they appear in the output, creating
differences between the output and ground truth. In addition to this, while sections of
the graph in the center and at the end nearly match the ground truth perfectly as seen
back in Figure 12 above, much of the early output graph does not, especially when many
of the different locations and entities are talked about but not physically present, such
as when Mufasa is teaching Simba. Both this and the musical numbers heavily decrease
the location and scene accuracy for an output that visually is rather close to the ground
truth.

### 4.5.1 Adherence to the Definition of Plot

As defined in section 2.1, plot requires (1) characters of volition, (2) events involving those characters, (3) representation on how
information spreads, (4) causation linking these aspects of plot, and (5) a full structure
of those links from the beginning to the end of the story.

The most obvious adherence to the definition is that the entities of the narrative flow
graph cover the 1st requirement. All plot-important characters and object are denoted
in the user-defined glossary, and as such, they are tracked. Not only are any events
involving at least one of those entities included in a scene vertex of the graph, but
any mention of that entity in conversation or description is also tracked. A character
might not be physically part of a scene, but if that character is discussed or described
(information shared), an edge representing that entity connects to that scene showing
that inclusion. This also means the narrative flow graph covers the 3rd requirement,
although there is currently no way to distinguish between a mention of an entity and
that entity being physically present.
Similar to satisfying the 1st requirement, the 2nd requirement is also covered by the detection of entities, but poorly. Every physical event of the story involving an entity is included in a scene vertex. The main issue here is that even those events that are not plot-important are also included. If a character sneezes for no apparent reason, that event will be included in a scene vertex because that character was directly referenced in the text. To fully satisfy the 2nd requirement, only those events that are vital to the story progression should be included, which the Plot Map System simply does not do.

The 4th requirement of causation is another that is covered but poorly. The links between scenes of the story are shown through the entities involved in each scene. As mentioned earlier, there is definitely a connection between scenes involving the same entity, meaning that the previous scene in some way “causes” the next where that entity next appears. The DAG’s dependency ordering of the graph illustrates these causal relationships. Where the current implementation lacks is that the reason for causal connection or what that specific causal connection is is not known, only showing that a causal connection exists. Defining the causal connection is a far more complicated task. Despite lacking in part, a causal structure is built mapping the story from its beginning to its end, satisfying the 5th and final requirement for plot according to the definition used in this research.

Following the requirements of plot detailed in section 2.1, the graph structure produced by the Plot Map System can be considered a partial representation the plot of the story.

5. Applications

The importance of this research comes in a number of areas. As mentioned in the introduction, analysis of literature is a rather time-dependent process. It requires one or more readers to personally parse through the text, and the time required for the analysis grows quickly the more pieces of literature that are included in the study, because each piece needs to be read and processed on its own. Major breakthroughs have come from the fields of NLP, digital humanities, and computational linguistics, but it yet remains very resource heavy depending on the task required. With regard to novels, simple statistical data can be gathered over a large numbers of books, but an in depth analysis of the narrative structure of the books is currently infeasible with large numbers of books. Due to this time constraint, literary analysis is primarily done on only a small collection of novels at a time for any single project. The two narrative visualizations presented in this paper give new visual and data representations of narrative that provide insight into the structure of the narrative visible at a glance without the need to read through the source text.

If there was a way to use NLP to automate the collection of complex information within the text, such as narrative structure and plot, literary analysis on a large scale could become possible. For example, Christopher Booker claims that there are only seven basic types of plot in fiction (Booker 2004). Could the analysis of the plot of tens or hundreds of thousands of novels support or disprove this claim? Could a newly published book be compared to all previously published works to both check for originality in narrative flow and detect potential plagiarism? Given multiple different discourses for a story, could the underlying story be constructed from the discourses? Could inconsistencies between discourses reveal through the narrative structure if one discourse is unreliable? In application, if these visualizations could be improved further, non-biased comparisons of different narrative discourses of the same story could be conducted to immediately see what scenes were excluded from one account and mentioned...
in another or where scenes differed between accounts. This is useful in news analysis and even crime analysis for testimonies and witness accounts, providing another tool to assist human analyzers.

Machine learning has been used to generate large volumes of text (Guo et al. 2017), extract information from text, and more, but most corpora of books solely include the text of the books and other external data such as author, publisher, and genre. Those corpora that do include detailed information on the narrative are created at great cost and are often much smaller than plain-text corpora. In order to use raw text as input to a machine learning system, researchers are often left to parse through the text themselves to find the specific data they need to use in their machine learning system or use the entire raw text with the lofty goal of letting the system identify those aspects of the text it needs, often with some human-in-the-loop guidance along the way; a monumental task. Having a method to extract plot from a text enables the ability to streamline the creation of a corpus of plot. From that corpus, machine learning can be used for tasks such as plot classification or potentially generating its own plot. When generating large volumes of text using existing methods, there is no plot structure to it unless it is pre-defined or human-directed. Some success has come from short text passages (Fan, Lewis, and Dauphin 2018), but for longer passages, a system pre-trained on hand-extracted plot-graphs is needed (Li et al. 2013). If a text generator could also learn how to weave its own plot, could a computationally creative system be developed that writes stories with long and intricate plot, such as novels, completely free of user oversight? If an author is stuck on how to continue the story or bridge two sections of the narrative, could a system trained on narrative structure be developed to provide suggestions on how to continue or structure the narrative going forward? Could machine learning be used to find what types of plot or narrative flow could lead to a best selling novel, training a the system on all the best selling novels over the past few decades? All of these questions and problems and more may be possible if the extraction of complex aspects of narrative, such as the structure or plot, could be automated.

6. Future Work

The Plot Map System is currently incomplete in that it is not fully autonomous. To enable full automation of the extraction of plot and narrative structure, future research needs to be conducted on the detection of plot-important entities in a story; this way they can be detected by the system itself instead of relying on a user-made glossary. This does not diminish the strength inherent in a user determining what entities they want the system to follow. Having an automation option only enables a wider application of the system to different tasks. Until the above goal is reached, this graph method of plot extraction and narrative visualization can be further refined to better represent the plot of the stories, including increasing the accuracy of location inference, scene divisions, determining which sections of the text are plot-important, and better handling of dialogue within the text to differentiate between when entities take physical action and when they are referenced in dialogue. Improvement in any of these areas should increase the quality of the plot maps produced by the Plot Map System.

7. Conclusion

Two different visualizations of plot and narrative flow have been presented. The Scatter-Plot of Entities, although lacking in its ability to fully represent the plot of a story, provides a new method to visualize entities within a story such as changes in scene
and story arc through the involvement of the major entities within the story. The Plot Map System better represents the plot of a story through a graph structure. It has proven capable of correctly visualizing sections of the narrative flow of a selection of stories of various formats, including short stories both professional and amateur as well as play and movie scripts, although it still falls short in correctly representing the entire plot map of any of these. Despite its faults, this method of plot and narrative representation and visualization is worth investigating and refining further to enable the automated extraction of these complex and vital aspects of story.

Acknowledgments
I would like to thank my advisor, Dr. Sean Warnick, for his guidance and assistance in this research. I would also like to thank Dr. Deryle Lonsdale for his instruction that inspired me to first begin my research with the creation of the Scatter-plot of Entities. Lastly I would like to give a special thanks to Alyssa Crezee, Caelen Miller, Maxwell Hamilton, Xin Zhou, Henry Ward, and Griffin Holt of my annotation team for their assistance with data preparation and creation of the ground truth data.

References


Naughton, Martina, Nicholas Kushmerick, and Joseph Carthy. 2006. Event extraction from heterogeneous news sources. In *proceedings of the AAAI workshop event extraction and synthesis*, pages 1–6.


Veselovsky, Alexander. 2015. From the introduction to historical poetics: Questions and answers (1894). *translated