Heroes and Villains: What A.I. Can Tell Us About Movies

Brahm Capoor
Department of Symbolic Systems
brahm@stanford.edu

Michael Troute
Department of Computer Science
mtroute@stanford.edu

Varun Nambikrishnan
Department of Computer Science
varun14@stanford.edu

Abstract

We present three distinct uses of Artificial Intelligence to gain insight into movie screenplays. First, we build models that classify characters as protagonists, antagonists, or neither, and evaluate their performance, both in classifying individual characters as potential protagonists or antagonists as well as in identifying the protagonist and antagonist of a given movie. Second, we cluster characters together using various sentiment scores to identify groupings of characters. We do this both for individual movies and all the movies in our dataset and explore interpretations for clusterings with different numbers of groups. Finally, we cluster entire movies by their emotional trajectories to identify archetypal stories and use a self-organizing map to visualize and verify these archetypes.

1 Introduction

When watching a movie for the first time, an avid movie-goer asks themselves several questions: who is the protagonist? Who is the antagonist? How are the supporting characters aligned (or not)? Is it a story they recognize, or something entirely new? These questions are often difficult to answer, and their answers are often colored by the biases of the movie-goer themselves.

Sharing a common love of film, we were interested in bringing A.I. techniques to bear on these questions. Given the screenplay of a film, we examine what sorts of insights we might be able to glean into how it works.

We use distinct techniques to answer three questions. First, we use supervised learning to try and identify the Protagonist and Antagonist of a film. Next, we cluster the characters of a film based on various sentiment scores and their relationship to the protagonist to identify different factions within the film. Finally, we find the emotional trajectories of films as the plot develops and cluster those trajectories to identify archetypical stories.

2 Literature Review

There is a relative sparsity of existing literature involving using AI techniques to gain insights into movie characters, their relationships, and the trajectories of their stories. A paper from the Austrian Research Institute for Artificial Intelligence describes the identification of character types from movie dialogues using linguistic analysis and communication characteristics [4]. The authors argue that the linguistic features of dialogue alone are not enough to comprehensively identify all character types, and thus they incorporate semantic graphs which take into account such information as the character's social role and linguistic markers of expressivity. In addition, they note that movies such as action movies most clearly exhibit common narrative properties.

The University of Vermont [3] used data mining and sentiment analysis on a corpus of classic novels from Project Gutenberg, and identified 6 key 'shapes of stories', or archetypes that represented all their novels.

As existing work has been focused primarily on gaining insights into literature, we see an opportunity to apply similar techniques to film and screenplays. Movies' relative simplicity when compared to literature both simplifies and complicates this application; while relationships and progressions in film tend to be simpler, there is also significantly less information from which to draw conclusions about them.

3 Task Definition

3.1 Detecting Protagonists and Antagonists

First, given the subset of labeled examples in our dataset, we apply supervised learning and sentiment analysis to learn models that predict, given a character, whether they are a protagonist or antagonist. This problem is fairly simple, in that the vast majority of film characters are not protagonists or antagonists, so attaining high accuracy is not particularly challenging. Much more difficult is the task of, given a film's screenplay as input, predicting *which* of the characters in the film is the protagonist and which is the antagonist.

Note that we define a film's protagonist loosely as the character around whom the film revolves, and with whom the audience most closely identifies; further, we define the antagonist as the character who stands in greatest opposition to the protagonist's goals.

3.2 Character Clustering

Second, we apply unsupervised learning and sentiment analysis to (a) group characters within the same film into certain factions or aligned groups, and (b) group characters across our entire dataset to identify common roles, or character tropes, in film. Examples of such tropes include mentor figures and romantic interests, among many others.

3.3 Shapes of Stories

Third, and finally, we seek discover recurring patterns in the emotional trajectories of the stories told by film. To do so, we apply sentiment analysis to on distinct sequences of dialogue from the screenplay to determine such a trajectory for each of the films in our dataset. Given these trajectories, we apply unsupervised learning to determine a set of archetypical stories. Once we've determined these archetypes, given a previously unseen film, we can quickly determine which archetype we believe it belongs to.

4 Data and Infrastructure

We use the Cornell Movie-Dialogs Corpus, [1] which contains 220,579 lines of dialogue spoken between 9,035 characters in 617 movies. The corpus also includes metadata such as film genre, character gender, and character position in movie credits.

We hand label protagonists and antagonists for a subset of the films in the dataset, in order to train our supervised learning models for protagonist and antagonist detection.

[1] available at https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

5 Approach

5.1 Detecting Protagonists and Antagonists

5.1.1 Oracle and Baseline

Our oracle for this prediction task is simply a human identifying both the protagonist and antagonist of a film; as such (and because there is no objective "ground truth" to speak of, but that's an entirely different issue), the oracle achieves perfect accuracy in both protagonist and antagonist identification, with the help of the occasional Wikipedia page.

Our baseline algorithm counts the number of times each character speaks in the film, and then considers the character with the most lines to be the protagonist, and the character with the second most lines to be the antagonist. Surprisingly, this approach is reasonably successful at identifying protagonists. It is able to identify the protagonist for 88% of films, failing mostly when the protagonist is the quiet, stoic type or the protagonists are a group of individuals who share the majority of the film's dialogue (as is the case in many ensemble films).

However, the baseline algorithm is far less successful at identifying antagonists: it is only correct 28% of the time. The baseline correctly identifies both protagonist and antagonist only 24% of the time.

This suggests that identifying antagonists is a significantly more difficult task than identifying protagonists. The baseline algorithm frequently misclassified important supporting characters (such as the aforementioned mentors and romantic interests) as antagonists, because their proximity to the protagonist tends to guarantee a high line count. The difficulty of identifying antagonists makes sense: it tends to be much easier to agree on a film's protagonist than its antagonist.

5.1.2 Protagonist Feature Extractor

To predict protagonists and antagonists, we built two feature extractors which are applied to the characters in our dataset. The first is the protagonist feature extractor, which includes the following features:

1. Number of lines spoken
2. Number of words spoken
3. Number of times spoken to
4. Whether any part of the character's name is in the title of the film
5. The position of the character in the film's credits
6. Whether the character is male
7. Whether the character is female

These features are fairly low-level (i.e. they do not take into account any semantic or sentimental knowledge of the film), but we found they led to high performance in our predictions. Note that Features 4, 6 and 7 are indicator features that are 1 when their corresponding condition is true and 0 otherwise. Note as well that features 6 and 7 are separated to accommodate characters that are not human or have no explicitly defined gender.

5.1.3 Antagonist Feature Extractor

In order to extract features for antagonist detection, we use all the same features as we do for protagonist detection, but given our identification of a protagonist, we are also able to use two higher-level features, namely:

1. Average sentiment of the lines they speak
2. Average sentiment of the lines the protagonist speaks towards them.

The combination of these higher-level features alongside the low-level features outlined in 5.1.2 are a rough approximation of the social and linguistic cues employed in [4] when classifying characters.

Sentiment information is found using NLTK's pre-trained 'Vader' sentiment analysis tool [1].

5.1.4 Detecting Protagonists and Antagonists

We train logistic regression models [2] on a training set representing 75% of the labelled films in our corpus. We assess these models in two ways. First, for every character in our test set, we classify the character as either a protagonist, an antagonist, or neither. Second, on a movie-by-movie basis, we identify the characters who has the highest probability of being the protagonist; given that choice as protagonist, we also identify a predicted antagonist. The following table describes our accuracy as it relates to each task:

Task	Train Acc.	Test Acc.
1	96.7%	95.8%
2	93.2%	93.3%
3	77.2%	74.1%
4	26.5%	33.3%

Here, Task 1 is the binary classification task to predict whether each character in the dataset is or is not a protagonist. Task 2 is the equivalent binary classification of characters as antagonists; note that it is dependent on the outcome of Task 1. Task 3 is identifying the correct protagonist from within the set of characters for each movie in the dataset; Task 4 is the equivalent task for the antagonist, and is dependent on Task 3.

The models are significantly more successful at tasks 1 and 2 than they are at tasks 3 and 4.

5.2 Character Clustering

In the second phase of our project, we apply k-means clustering to split characters into factions with the aim of discovering groupings of similar characters. Each character is represented by their overall average sentiment as well as the protagonist's sentiment towards them (see section 5.1.3). We run k-means individually on each movie, as well as a global k-means on all characters across all films.

5.2.1 Individual Movie Clustering

Initially, we cluster on every character in a movie. However, we find that including the classified protagonist in the clustering oftentimes results in either the protagonist being placed in their own cluster, or heavily distorting another cluster that would have been much tighter without them. Additionally, since a protagonist doesn't have an explicit particular sentiment towards themselves, including the protagonist in this clustering is somewhat nonsensical. Thus, for each movie, we cluster on every character in the movie except for the classified protagonist. We run 2,3,4, and 5-means on each movie, as using more than $K = 5$ clusters would make it difficult to attach any semantic meaning to the clusters. Examples of these clusterings can be found in section 6.2.2.

5.2.2 Global Character Clustering

In addition to clustering locally on each individual movie, we also apply clustering to the set of all characters in the corpus. We once again filter out the character classified as protagonists. This global clustering aims to identify groups of characters across different movies which represent archetypes or tropes, such as buddies, villains, romantic interests, and mentors. We again

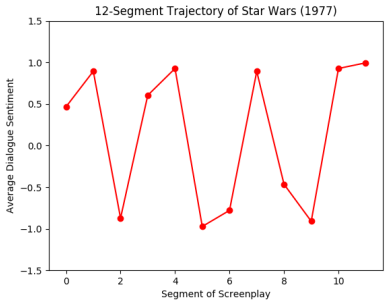


Figure 1: The emotional trajectory of *Star Wars*

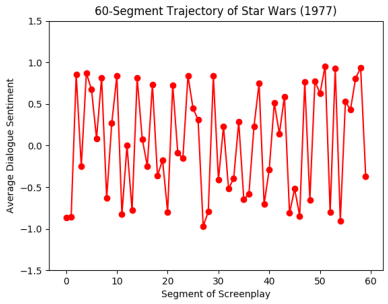


Figure 2: The noisy, detailed emotional trajectory of *Star Wars*

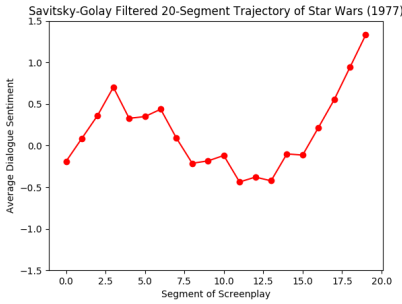


Figure 3: The emotional trajectory of *Star Wars*, smoothed using Savitsky-Golay filtering

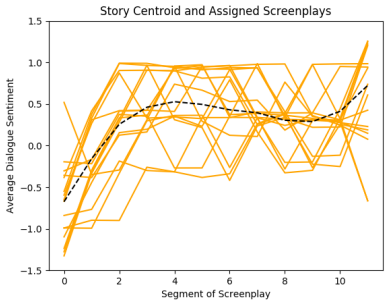


Figure 4: The Hero's Journey

ran 2,3,4, and 5-means clustering on the entire corpus of characters. Analysis of these clusterings can be found in section 6.2.2.

5.3 Shapes of Stories

In the third phase of our project, we use unsupervised learning to find patterns in the emotional trajectories of the scripts in our dataset. To obtain an emotional trajectory for a film, we first concatenate all the tokens of all the lines spoken in the film into a single ordered list. We then split this list into N equal-length "segments". Using NLTK's pre-trained 'Vader' sentiment analysis module [1], we can then determine a sentiment score for each of these segments; taken together, these scores define a rough emotional trajectory for the film. Figures 1 and 2 show the emotional trajectory of the original *Star Wars* (1977) for 12 and 60 segments respectively.

Both of these figures, especially Figure 2, are fairly noisy; as N increases, individual segments become smaller and smaller, and tend to produce highly volatile trajectories. To address this, we apply a Savitsky-Golay filter[2] to the trajectories. This produces the the smoothed version of *Star Wars*' 20-segment trajectory visible in Figure 3.

Once we have obtained smoothed trajectories for each of the films in our dataset, we can perform K-means clustering on the trajectories (as points in N-dimensional space) in an attempt to cluster stories that follow the same arc or have the same general shape. Figures 4 and 5 show two of the centroids, and the stories assigned to them, learned with $K = 7$ and $N = 12$.

We call the first of these story archetypes (Figure 4) the Hero's Journey, and the second (Figure 5) the Redemption Story. Graphs and names for all these archetypes are included in

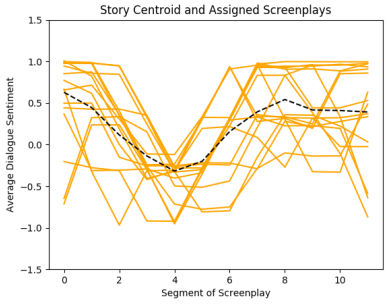


Figure 5: The Redemption Story

[2]We also attempted other smoothing techniques, such as a simple sliding filter, but they tended to yield centroids that were not as distinct as Savitsky-Golay. Tuning the window size and polynomial order for Savitsky-Golay was another important process that we omitted from this report.

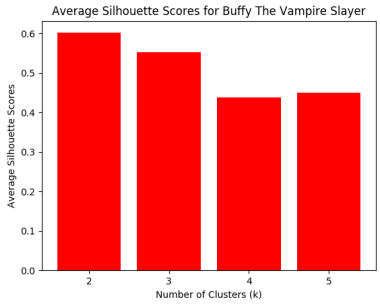


Figure 6: Silhouette scoring for clustering on *Buffy the Vampire Slayer*

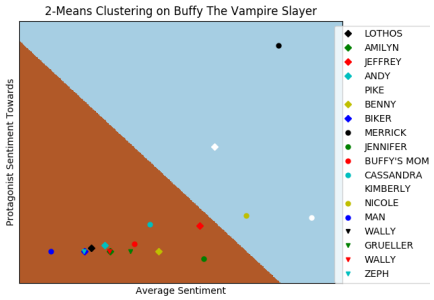


Figure 7: Visualization of 2-means clustering for *Buffy the Vampire Slayer*

Appendix A. The author Kurt Vonnegut suggested in his masters thesis in anthropology that all stories have 6 distinct shapes, a hypothesis later reaffirmed by the University of Vermont's research[3]. We identify these same 6 shapes of stories in our films, alongside a seventh cluster of films that do not fit particularly well into any cluster.[3] This last cluster is likely the product of films having cinematic narrative devices that are impossible to capture from text alone and so we are unable to capture to which archetype these films belong. It is easy to read such meaning into story centroids produced by any combination of K and N, so choosing these values carefully is of paramount importance, as will be discussed further in the next section.

6 Experiments and Error Analysis

6.1 Detecting Protagonists and Antagonists

As reported in section 5.1.4, our logistic regression model does not perform ideally at detecting protagonists and antagonists within a given movie. This being the case, we see three future avenues for improving the performance of our model. First, we could use domain knowledge to generate more features from our corpus; second, we could label more films from our dataset or obtain a larger dataset to better train the model; third, we could implement another model entirely, such as a neural network or support vector classifier. While each of these approaches is promising in its way, we elected to focus our remaining time on further exploring Character Clustering and especially Shapes of Stories, as these areas allow us to apply artificial intelligence techniques beyond the supervised learning context.

[3]Graphs of each of the generated clusters, along with a subset of the movies assigned to them, can be found in Appendix A.

6.2 Character Clustering

6.2.1 Silhouette Scoring

We use Silhouette Scoring as a metric for determining the optimal value of K in both Character Clustering and Shapes of Stories. The Silhouette Score is a metric that measures how similar an object is to its own cluster as compared to other clusters, effectively gauging how well a point fits in its own cluster. The score ranges from -1 to 1 where a high value indicates a data point is well matched to its own cluster and poorly matched to other clusters.

6.2.2 Different K for Different Movies

For each possible value of K, we calculate the average Silhouette Scores and choose the value of K that yields the maximum as our number of means. We found that movies where the protagonists and antagonists are unambiguous (as they are in typical action, thriller, and horror films), a lower number of clusters led to a higher average Silhouette Score. Buffy the Vampire Slayer, for example, a movie whose entire premise can be expressed by its title, has the best average silhouette score for $K = 2$, as shown in Figure 6. The resulting clustering is shown in Figure 7.

Conversely, in movies where relationships are not as simple or clearly defined–as is often the case in films with several sets of characters with distinct interests and alignments, increasing the number of clusters can improve the silhouette score. For example, Star Trek III: The Search for Spock, receives the best silhouette score for $K = 5$, as shown in Figure 8. The resulting character clustering, shown in Figure 9, reflects the complexity of a film with multiple star-faring factions.

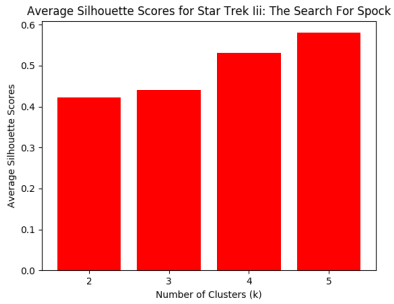


Figure 8: For *Star Trek III*, $K = 5$ produces the best score

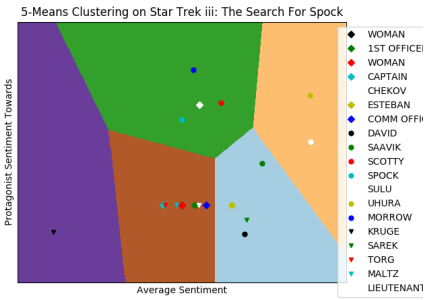


Figure 9: Visualization of 5-means clustering for *Star Trek III*

6.2.3 Optimal Global K

The average silhouette scores for clustering on all the characters in our corpus with different values of K are shown in Figure 10. We see that overall, the average silhouette scores trend downward as the number of clusters increases. This is a good indication that the data is not well separable into a large number of clusters. That being the case, it is still possible to assign semantic meaning to clusterings with different values for K. Clustering with $K = 2$, as shown in Figure 11, produces one cluster of mentors and other important supporting characters (for example, Gordon Gekko in *Wall Street*, Merrick in *Buffy the Vampire Slayer*, and Han Solo in *Star Wars*), and a second cluster of antagonists, minor, and one-string characters.

While $K = 2$ produces the highest silhouette score, there are viable interpretations for other clusterings as well. Consider clustering with 4-means as shown in Figure 12. One possible interpretation is that the four clusters represent enemies, minor characters, supporting characters, and character foils.

While Silhouette Scoring is a valuable metric for choosing cluster sizes, we see there are viable interpretations of results where K may not have the best Silhouette Score.

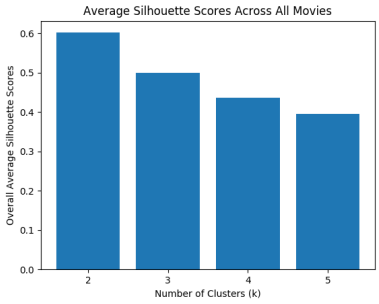


Figure 10: Silhouette scores across all characters in the corpus decrease as K increases

Figure 11: Cluster centroids shown in white; points in the brown region tend to be supporting characters, points in the blue region tend to be enemies and minor characters

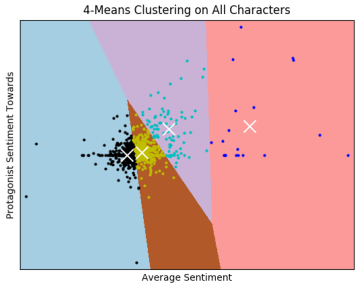


Figure 12: Points in the blue region tend to be enemies, points in the brown region tend to be minor characters, points in the purple region tend to be supporting characters, and points in the pink region tend to be foils.

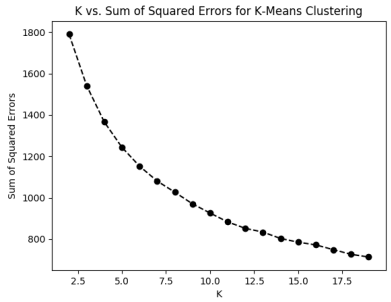


Figure 13: Note the absence of a clear kink in the plot

6.3 Shapes of Stories

6.3.1 Choosing N and K

As previously mentioned, there are many combinations of values for N and K which produce clusters that appear to have semantic meaning. Therefore, it is important that implement some methodology for comparing the outcomes produced by clustering for different such combinations. We explore two approaches. First, we use the "elbow method", plotting increasing K against sum of squared error; a clear "elbow" or kink in the resulting graph represents a value of K such that further increasing K brings sharply decreasing rewards. Figure 13 is an example of one such graph.

The lack of any distinct kink suggests that our sentiment trajectory data are not easily separable; therefore, increasing the number of clusters steadily decreases the sum of squared errors.

Second, we apply silhouette scoring, as described in section 6.2.1. Figure 14 plots K against silhouette score for different values of N. Good values for K and N, in light of our task definition, are values which compromise between relatively high silhouette scores, ease of (human) recognition of the resulting archetypes, and middling segment sizes, such that the larger emotional trajectory of the films remains visible and is not eliminated by filtering.

We concluded that $K = 7$ and $N = 12$ was such a compromise, though it should be noted again that this is not the only viable interpretation of our results.

6.3.2 Self-Organizing Maps for Visualization and Verification

In order to assess our choices of N and K and verify that they were sensible, we used a self-organizing map (SOM), an unsupervised neural network that reduces points in a high-dimensional space to produce a low-dimensional representation of them. This representation

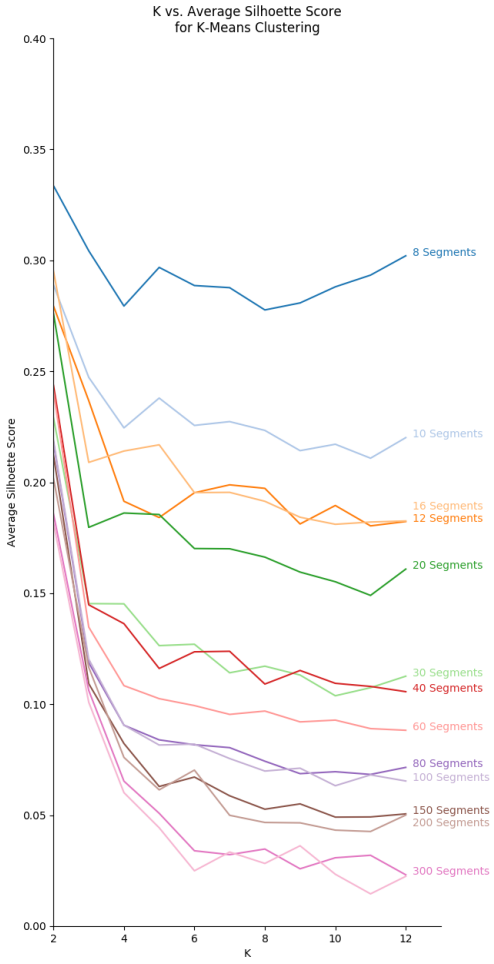


Figure 14: Visualization of the interplay between K, N, and silhouette score

Figure 15: A self organizing map provides visual insight into the similarities between clusters

essentially maps points in a higher dimension $\mathbb{R}^N$ (in our case, $\mathbb{R}^{12}$) to discrete points in a lower dimension (in our case, $\mathbb{R}^2$). One key property of an SOM is the fact that distances in the lower-dimensional representation reflect topological similarity: that is, points that are close together in $\mathbb{R}^N$ are mapped to points that are close together in $\mathbb{R}^2$.

We train an SOM on the smoothed emotional trajectories of each of the films in our corpus, keeping in mind that films with similar emotional trajectories (represented as points in $\mathbb{R}^{12}$) will be mapped to nearby points in $\mathbb{R}^2$. We mark each point in $\mathbb{R}^2$ with a circle whose size represents the number of movies that map to it and a color that represents the average archetype index (as determined by our K-Means algorithm) of all the movies that map to it. This produces the image in Figure 14.[4]

The numbers inside each circle represent the archetype to which the majority of the films that map to the point are assigned. The SOM shows fairly well-defined clusters of particular colors, implying that movies of the same archetype are close together in their 12-dimensional space as well. This lends credibility to our clustering of emotional trajectories, since trajectories of the same archetype demonstrate strong similarities to each other, and also provides some interesting visual insight into which clusters are most similar and most distinct.

7 Conclusion

Ultimately, our results amount to interesting analytical tools. They are more interesting as a means to examine and verify our preconceived assumptions about film than to generate wholly new conclusions.

Viewing our work through that lens provides three interesting takeaways. First, we learn that it is easier to tell that an individual character is a hero or a villain than it is to look a group of characters and conclude who among them rises above and beyond. We feel this ambiguity in real life, when each character on their own is easy to consider significant or otherwise, but when considered collectively, it is harder to discriminate between them. Second, while there are a plethora of ways to divvy up the characters in a film into various factions, some have more distinct merit than others and provide clearer perspectives into the relationships the script is founded upon. Finally, despite their seemingly inexhaustible variety, the stories humans tell – no matter the medium – almost always are part of a larger collective of similar stories. That said, films, as visual media, have at their disposal narrative tools that rise above the words their characters speak and tell the same story in a completely different way. In the future, it would be interesting to attempt to use computer vision techniques to include visual storytelling techniques in some sort of similar clustering analysis, to discover whether patterns found in visual storytelling mimic in some way those found in verbal storytelling.

[4] A much larger version of the same image can be found in Appendix B.

References

[1] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python.* O'Reilly Media, 2009.

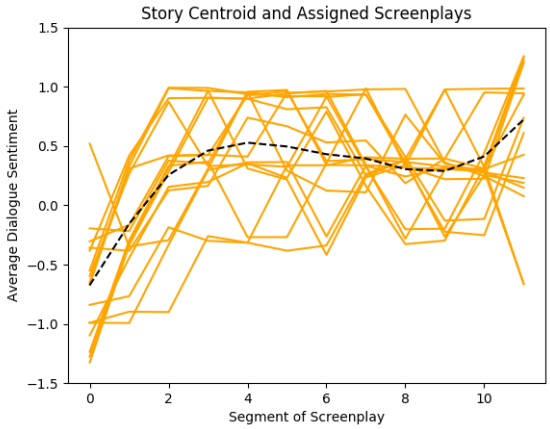
[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] A. J. Reagan, L. Mitchell, D. Kiley, C. M. Danforth, and P. S. Dodds. The emotional arcs of stories are dominated by six basic shapes. *CoRR*, abs/1606.07772, 2016.

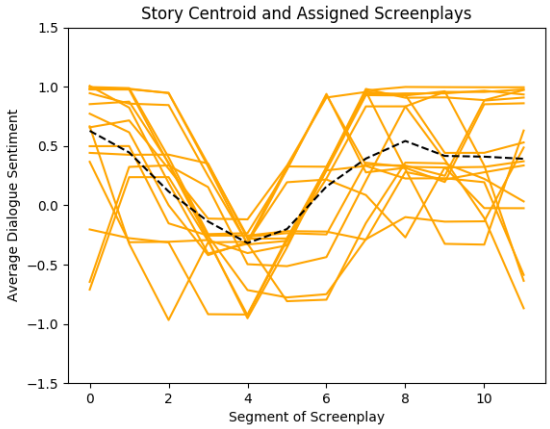
[4] M. Skowron, M. Trapp, S. Payr, and R. Trappl. Automatic identification of character types from film dialogs. *Applied Artificial Intelligence*, 30(10):942–973, 2016.

Appendix A: Story Shapes for $K = 7$, $N = 12$

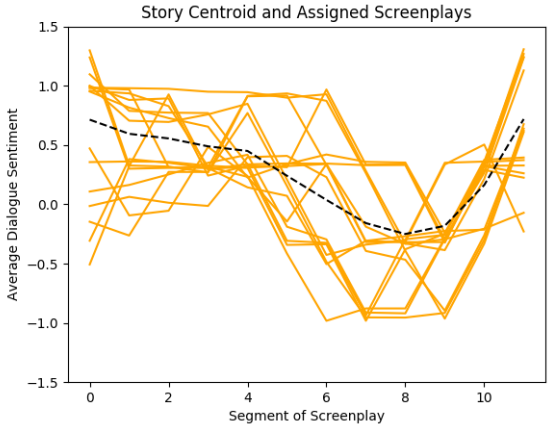
Hero's Journey



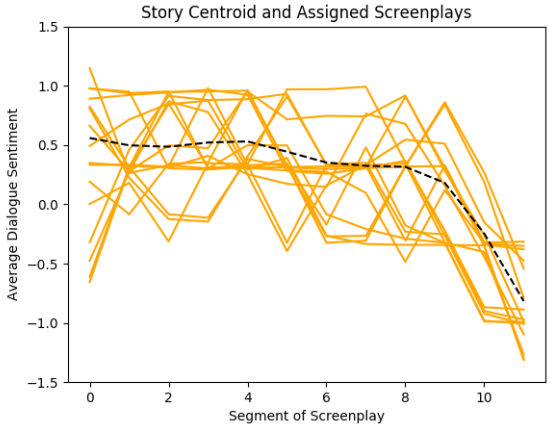
Redemption



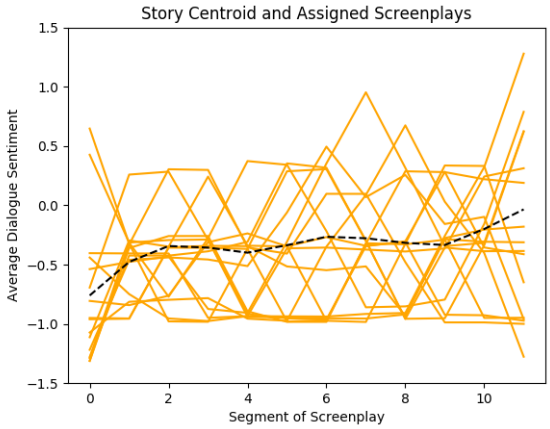
Man in Hole



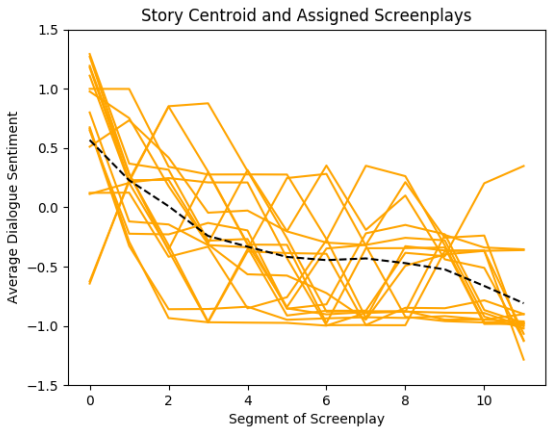
Breakdown



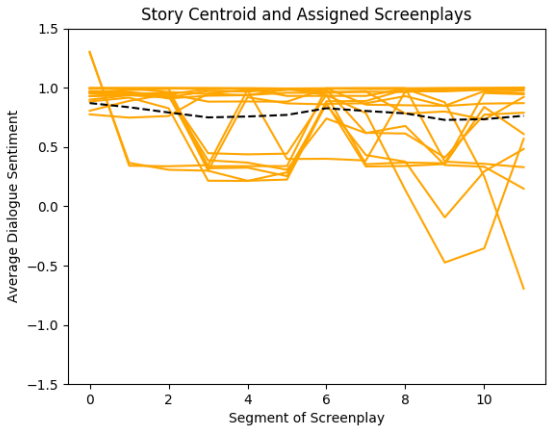
Creation



Tragedy



Non-Clustered



8 Appendix B: Larger Self-Organizing Map