Extraction and Analysis of Character Interaction Networks From Plays and Movies

Sebastian Gil, Laney Kuenzel, and Caroline Suen
CS 224W Final Project Report
Sunday, December 11, 2011

1 Introduction

Due to recent efforts to digitize literary works, researchers have been able to perform large-scale analyses of millions of texts and reach meaningful conclusions about literature, language, and culture [1]. The vast majority of such work has focused on analyzing literature at the level of individual words. In an early landmark example, Mosteller and Wallace [2] used statistical analysis of word frequencies to determine that, with very high probability, the twelve unattributed Federalist Papers were written by James Madison. More recently, Michel et al. [1] made fascinating discoveries about collective memory, the evolution of grammar, the dynamics of fame, and more by examining word frequencies in millions of books published over the past two centuries. While this statistical approach is clearly powerful, it has certain drawbacks. By considering only coarse-grain properties like word frequencies, these analyses completely ignore the subtleties of the literary works in question, reducing complex texts to bags of words.

Compared to computer scientists, literary theorists typically take a very different approach to the task of analyzing literature. Rather than focusing on frequencies and statistics, they tend to perform in-depth qualitative studies examining the intricacies of plot structure and character interactions. Unfortunately, while this form of detailed analysis allows for deep conclusions to be drawn about literary works, it does not scale well to large numbers of texts due to the significant amount of time required for a human to read, understand, and thoroughly analyze a piece of literature.

In this project, we combine the two approaches to literary analysis (that of a computer scientist and that of a literary scholar), allowing us to benefit from the advantages of both. More specifically, we develop and apply methods for automatically extracting character interaction networks from works of entertainment and use the properties of the resulting networks to draw conclusions about the works at hand.

There are three main components of our project: (1) extracting character interaction networks in the form of weighted graphs from our chosen works, (2) computing informative properties (e.g., clustering coefficient) of the resulting networks, and (3) using those properties to answer broad questions about the works (e.g., whether different genres or media types are characterized by distinctive types of interaction networks) by constructing logistic regression and decision tree classifiers. We discuss all three steps, as well as our data sources, in greater detail below. First, we briefly survey some related work.

2 Related Work

As described in Section 1, most computational literary analysis has been at the word level. There are, however, several exceptions to this generalization. Most notably, Elson et al. [3] extracted social networks from sixty 19th-century literary works using dialogue interactions. They used techniques from natural language processing and machine learning to determine all likely names for each individual character, identify quoted speech, and attribute each quote to a character. Using the resulting data, Elson et al. constructed a conversational network with vertices corresponding to characters and weighted edges corresponding to the amount of conversational interaction. They then examined various properties of these networks (e.g., density, average degree, and number of triangles) and tested whether these properties were correlated with general aspects of the works, such as setting (urban vs. rural) and number of characters. They showed that their analysis contradicted two prevailing hypotheses held by 19th-century literature scholars. Their work was new and creative, and it allowed them to make interesting discoveries about a particular genre.

Several other researchers have also used network theory to analyze individual texts or small groups of texts. For instance, Moretti [4] used character interaction networks in an analysis of Hamlet, and Rydberg-Cox [5] created an application to visualize and explore a social network built from Greek tragedies. Stiller and Hudson [6] found that character networks in ten Shakespeare plays exhibited small-world properties, and Alberich et al. [7] discovered that the network of characters in Marvel Comics is quite similar to various real-world networks. Each of these previous studies has been relatively narrow in focus, leading to valuable discoveries about a small number of texts.

For works of entertainment other than literary texts, C.-Y. Weng et al. [8] recently proposed a method for extracting social networks from movies or TV shows based on co-occurrence in scenes. They applied this method to several movies and demonstrated the usefulness of the resulting networks in identifying lead roles, finding communities of characters, and performing story segmentation.

Overall, the previous work has primarily focused on using character interaction networks to improve understanding of individual texts or movies. We feel that humans can already do a very good job—better than computers—of analyzing small collections of works; the main limitation of humans is that we do not have the brainpower to simultaneously analyze and compare hundreds or thousands of works. For this reason, we are interested in conducting a large-scale study of the character interaction networks in many diverse works of entertainment. Our goal is not to examine the literature from a specific time period or the plot of a particular film but rather to discover sweeping trends in literature and movies across genres and over time.

3 Methods

3.1 Building Networks

3.1.1 Data Collection

In our project, we chose to focus on play and movie scripts because their structured format is well suited for systematically detecting interactions between characters. Our first step was to identify various freely available online sources for play and movie scripts. We used The Internet Movie Script Database [9]—an extensive script database for numerous popular movies—for all our movie scripts. For plays, we were not able to find a single exhaustive source, so we used multiple sources including Project Gutenberg [10], The Complete Works of William Shakespeare hosted at MIT [11], EOneill eText Archive [12], Read Print [13], and The EServer Drama Collection [14].

After identifying data sources, we developed methods to extract scripts from the sites and transform them into a standardized text format suitable for character interaction extraction. First, we investigated Project Gutenberg, a free online database containing over 36,000 literary works, all accessible from its

site gutenberg.org. Here, we expected to find many plays we could use in our analysis. However, the site does not distinguish between plays and novels, and it only supports browsing by author and title. In order to circumvent this limitation, we constructed a list of plays we hoped to find in their selection. Project Gutenberg's URLs are standardized and predictable so that given the author and title of a play, one can easily navigate to the correct browsing page that contains the play script. We thus constructed Python scripts to automate this process. After retrieving the HTML of the browsing page, we utilized regular expression matching to extract the work's identification number, which allowed us to retrieve the full text of the play in HTML format. More regular expression matching and replacements yielded the play in plain text format, suitable for character network interaction extraction.

We employed a similar technique on The Internet Movie Script Database, except that their browsing pages allowed us to extract all the movie scripts in their collection. The rest of the sites proceeded similarly, where a single index page allowed us to access their entire play collection (with a few differences in format). In total, we extracted 190 plays and 951 move scripts.

After extracting the scripts, we needed to collect additional metadata on them for use in our classifiers. To gather this information on movie scripts, we used The Internet Movie Script Database and Rotten Tomatoes [15]. The Internet Movie Script Database contained easily parsed release dates and genres for all movies, and we used the Rotten Tomatoes API to obtain the MPAA rating, audience score, and critic score for each of our selected movies. To gather additional information on play scripts, we turned to Wikipedia. We again constructed Python scripts to search for the relevant play on Wikipedia and used regular expression matching to extract the play release date and author after performing a query for the requested play. We entered some of this information manually when regular expression matching failed, followed by a review of the metadata for familiar works to verify the success of our methods.

3.1.2 Information Extraction

After amassing a collection of play and movie scripts, we found that many of the scripts were not in consistent formats. To ease the process of network extraction, we decided to convert all plays and movie scripts into a standardized intermediate format in which (i) each line of dialogue is represented by a single line of text in the form "Name - x," where Name is the name of the speaking character and x is the number of words spoken, and (ii) empty lines indicate scene breaks.

To obtain this format, we again used regular expression matching in Python to parse scripts and determine whether individual lines contained a scene or act break, a speaker name, dialogue, character actions, or a mix of all these options. Unfortunately, different movies and different plays often used different formats to display their information, and sometimes there were inconsistencies even within the plays and movies themselves. Such inconsistencies proved to be the most difficult obstacle to overcome in the information extraction process.

We wrote bash scripts to parse mass amounts of plays and movies at once, using different regular expression formats, and determined which parsed outputs were successful based on file output size and number of scene breaks found. Using one catch-all method of parsing would have been impossible, because what one source used to denote a speaker name may have been used by another source to denote spoken dialogue, for example.

To address the issue of inconsistencies within individual scripts, such as when an action such as "fade in" appeared in the same format as that used for speaker names, we created a "blacklist" of scene command regular expressions to ignore. We also disregarded all speakers with fewer than five total lines. These two strategies combined resulted in much cleaner and more consistent networks. Overall, we successfully converted 580 movie scripts and 173 plays to our standardized intermediate format.

3.1.3 Network Extraction

Once we had transformed all of these works into the intermediate format, we extracted our character interaction networks. We experimented with four extraction algorithms. The first one, the approach used by Weng et al. [8], defined the interaction score for two characters as the number of scenes in which both characters appear. Our second algorithm was an extension of the first to incorporate the number of lines each character speaks in each scene. These first two algorithms were attractive in their straightforwardness, but we found that they had serious shortcomings. Most significantly, we discovered by examining our data that many of our plays and movie scripts had long scenes, leading to falsely high interaction scores when two characters speak lines in different parts of the same scene.

To address this issue, we developed a second approach to network extraction which we refer to as the *Closeness* approach. This algorithm considers an interaction to have occurred between two characters whenever the characters speak nearby lines in the same scene. The precise definition of "nearby" depends on a parameter m; whenever characters i and j speak lines in the same scene that are within m lines of one another, we increment the interaction score w_{ij}. The amount by which we increment the score decreases linearly with the number of intervening lines. Our fourth algorithm was a variant of the *Closeness* approach which weighted an interaction between two characters by the total number of words exchanged in the interaction.

After examining and visualizing the results of the various algorithms for familiar works, we determined that the *Closeness* method with parameter $m = 10$ and without word count weighting yielded the best interaction networks. Thus, we chose to use that method to obtain our networks for the analysis described below.

3.2 Property Calculation

For each of our character interaction networks, we computed the following network properties:

- **Average clustering coefficient (unweighted and weighted)**. To calculate the unweighted average clustering coefficient, we disregarded our edge weights and used the standard definition of clustering coefficient. To compute the weighted version, we used the generalization of clustering coefficient to weighted networks described by Onnela et al. [16]:

$$C_i = \frac{1}{k_i(k_i-1)} \sum_{j,k} \left(\frac{w_{ij}}{w_{\max}} * \frac{w_{ik}}{w_{\max}} * \frac{w_{jk}}{w_{\max}} \right)^{\frac{1}{3}} \tag{1}$$

  where k_i is the degree of node i, w_{ij} is the weight of the edge between nodes i and j, and $w_{\max}$ is the maximum edge weight in the network.

- **Single character centrality**. To quantify how much the work focuses on a single character above all others, we calculated the following property, where s_i denotes the weighted degree of node i and the next-max operator gives the second highest in a group of values:

$$SCC = \frac{\max_i(s_i) - \text{next-max}_i(s_i)}{\sum_i s_i} \tag{2}$$

- **Single relationship centrality**. To quantify how much the work focuses on a single relationship above all others, we calculated the following property, where w_{ij} again denotes the weight of the edge between nodes i and j:

$$SRC = \frac{\max_{i,j}(w_{i,j}) - \text{next-max}_{i,j}(w_{i,j})}{\sum_{i,j} w_{i,j}} \tag{3}$$

- **Top character weight variance**. We calculated the variance of the top 10 normalized weighted node degrees to quantify whether the work has a large group of characters with similar prominence (low variance) or a few main characters and other less important roles (high variance).
- **Top relationship strength variance**. Likewise, we calculated the variance of the top 10 normalized edge weights to quantify whether the work focuses roughly equally on many relationships (low variance) or emphasizes a few relationships (high variance).
- **Entropy of node degrees (unweighted and weighted) and edge weights**. As alternative approaches to quantifying the spread in the distribution of character and relationship importances, we calculated the entropy of the bucketed, normalized distributions of unweighted node degrees, weighted node degrees, and edge weights.
- **Mean and variance of top character relationship strengths**. These two properties attempt to quantify whether the work has one or several main storylines. We identified the top 5 characters based on weighted node degree, and then found the 10 edge weights between every pair of these characters. We normalized these weights by the total weight in the network and then computed mean and variance. We expected that works with several storylines would have important characters who do not interact much, leading to low mean and high variance for these weights.
- **Percentage of existing edges**. We computed the fraction of character pairs connected by an edge in the network as another simpler attempt to quantify whether there are many characters who do not interact with one another, suggesting the existence of multiple storylines.
- **Betweenness centrality: maximum, difference, and entropy**. We let the inverse of our edge weights represent a "distance" in our networks and computed the normalized betweenness centrality of each node. We then used as features the maximum betweenness centrality, the difference between the top two betweenness centralities, and the entropy of the bucketed distribution of betweenness centralities for all nodes.
- **Number of characters**. Our final feature was the number of characters in the network.

3.3 Classification

The central piece of our work was to use our network properties as features to build classifiers to distinguish our works based on aspects such as media type, date, popularity, and genre. More specifically, we attempted the following binary classification tasks:

- **Media type:** To classify works based on whether they are plays or movies (173 and 580 examples, respectively).
- **Date of movie:** To classify movies based on whether they were released before or after 2000 (344 and 235 examples, respectively).
- **Date of play:** To classify plays based on whether they were published before or after 1800 (64 and 109 examples).
- **MPAA rating:** To classify movies based on their MPAA ratings. We used two different splits: G/PG/PG-13 movies versus R movies (210 and 338 examples, respectively) and G/PG movies versus PG-13/R movies (57 and 491 examples, respectively).
- **Audience and critic ratings:** To classify movies based on how they were rated by audience members and movie critics. For both types of ratings, we divided the set of movies into equally-sized "above average" and "below average" classes by splitting around the median score.

- **Single genre:** To classify movies based on whether or not they fall under a certain genre.
- **Between genres:** To classify movies based on whether they fall under a first genre or a second genre. Most movies had multiple genres, and we excluded movies from this between-genre classification if they fell under both of the genres in question. Thus, the first class consisted of the movies in the first genre but not the second, and vice versa for the second class.
- **Author:** To classify plays based on which of two authors they have. There were only three authors for whom we had a significant number of plays: William Shakespeare (31 examples), George Bernard Shaw (18 examples), and John Galsworthy (11 examples).

For these tasks, we built two types of classifiers: logistic regression classifiers and decision trees. We chose these classifier types because they both enable us to not only make black box class predictions and compute performance metrics but also to understand how the features are being used to arrive at those predictions. More specifically, we can examine the feature weights of logistic regression classifiers to understand which features are most important, and we can examine the actual decision tree built to gain insight into feature interaction and relevance.

We now describe the procedure used for each of our classification tasks. Throughout these steps, we used Orange, an open-source Python library for data visualization and analysis. We first normalized all features to have mean zero and unit variance so that we could draw meaningful conclusions from the logistic regressions feature weights. We then split the dataset into a training set (consisting of 80% of the examples) and a testing set (consisting of 20% of the examples). For logistic regression, we sorted the features based on estimated quality according to the Relief algorithm [17] and defined a parameter k such that our classifier uses the top k features. We performed 5-fold cross-validation on the training set to select the optimal value for k, choosing the value which maximized area under the ROC curve (AUC). Similarly, when learning our decision trees, we performed 5-fold cross-validation on the training set to select the optimal values for two parameters ("minimum examples" and "maximum majority") that determine when the inductive process of tree-building stops. For the play author classification tasks, we used leave-one-out cross-validation rather than 5-fold cross-validation to select the optimal parameters due to the small number of examples.

Once we had determined parameter values, we trained our classifier on the entire training set and then tested it on the test set. Because our two classes did not always have the same number of examples, the classification accuracies were sometimes misleadingly high even for poor classifiers. Thus, we chose to use AUC as our primary performance metric.

4 Results and Findings

4.1 Raw Results

We found that the logistic regression classifiers had higher AUCs than the corresponding decision tree classifiers for 26 of the 35 classification tasks we tried. Furthermore, for 8 of the 9 tasks for which decision trees performed better, both types of classifiers performed relatively poorly (with AUCs less than 0.65), making small differences in performance largely irrelevant. We observed that the decision trees consistently had very high AUCs (around 0.8 or 0.9) on their training sets even when their testing AUCs were low. This result suggests that the trees suffered from overfitting despite our attempts to avoid that phenomenon by performing cross-validation for parameter selection. On the other hand, the testing AUCs for the logistic regression classifiers were usually much closer to their training AUCs, suggesting that we successfully avoided overfitting by performing cross-validation to select the optimal number of features. For these reasons, we chose to focus in this section on the results of our logistic regression classifiers and to only use the decision trees as an additional means of gaining intuition for the role of our features in classification.

Task	AUC
Movie vs. Play	0.892
Play: pre-1800 vs. post-1800	0.776
Movie: pre-2000 vs. post-2000	0.479
Movie: G/PG vs. PG-13/R	0.594
Movie: G/PG/PG-13 vs. R	0.538
Movie: Audience good vs. bad rating	0.449
Movie: Critic good vs. bad rating	0.468
Play: Shakespeare vs. Shaw	1.000
Play: Shakespeare vs. Galsworthy	0.929
Play: Shaw vs. Galsworthy	0.750

Table 1: Logistic regression classifier AUCs for various classification tasks

	Comedy	Romance	Drama	Action	Horror	Thriller	Crime
Comedy	0.690	0.320	0.632	0.773	0.825	0.650	0.573
Romance	0.320	0.565	—	0.561	0.682	0.614	0.646
Drama	0.632	—	0.576	0.721	0.667	0.587	0.692
Action	0.773	0.561	0.721	0.662	0.643	0.640	0.563
Horror	0.825	0.682	0.667	0.643	0.660	—	0.721
Thriller	0.650	0.614	0.587	0.640	—	0.527	0.622
Crime	0.573	0.646	0.692	0.563	0.721	0.622	0.454

Table 2: Logistic regression classifier AUCs for genre-related classification tasks

Our results are shown in Tables 1 and 2 below. Table 2 is organized with genres as rows and columns. The results of the single-genre classification tasks (i.e., whether or not a movie has the particular genre) appear along the diagonal. The results of the between-genre classification tasks (i.e., whether a movie belongs to the first or second genre) appear in the table cell with one genre as the row and the other as the column. (Note that each between-genre classification result appears twice in the table, once above the diagonal and once below, for ease of comparing results). Certain cells contain dashes because the large overlap between the two genres meant that we did not have enough examples of movies falling into one genre but not the other.

4.2 Analysis

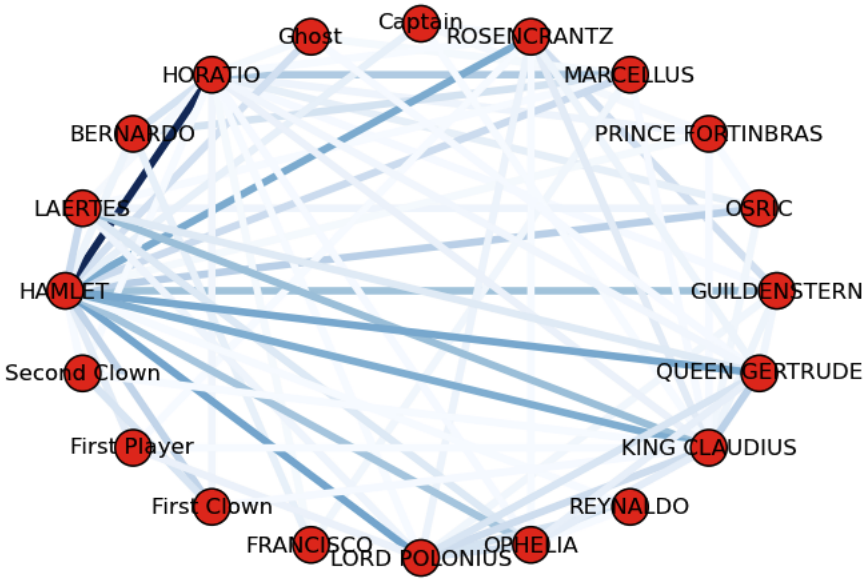
4.2.1 Feature weight interpretation

As mentioned above, we normalized our feature values to have mean zero and unit variance so that we could draw valid conclusions by considering the weights learned by our logistic regression classifiers. More specifically, features which are important for the classifier receive high-magnitude feature weights. Furthermore, features with higher values for the first class typically receive negative weights whereas features with higher values for the second class receive positive weights. We make use of these facts below as we analyze our classifiers.

4.2.2 Media type classifier

We were very successful in classifying works based on their media type: play versus movie. Examining the feature weights learned by our classifier led us to some interesting conclusions about what distinguishes the interaction networks of plays and movies. For instance, we found that plays are characterized by high top character relationship variance, high single character centrality, and low top character weight variance relative to plays. Combining these observations, it appears that a typical play has one clear-cut most important character as well as many supporting characters of roughly equal importance who participate in several distinct storylines. As an example, consider *Hamlet*, which has the interaction network displayed in Figure 1. Hamlet is clearly the main character, and there are many other important characters such as the queen, Ophelia, Rosencrantz, and Horatio who interact with Hamlet but not very much with one another.

Figure 1: Interaction network for *Hamlet*



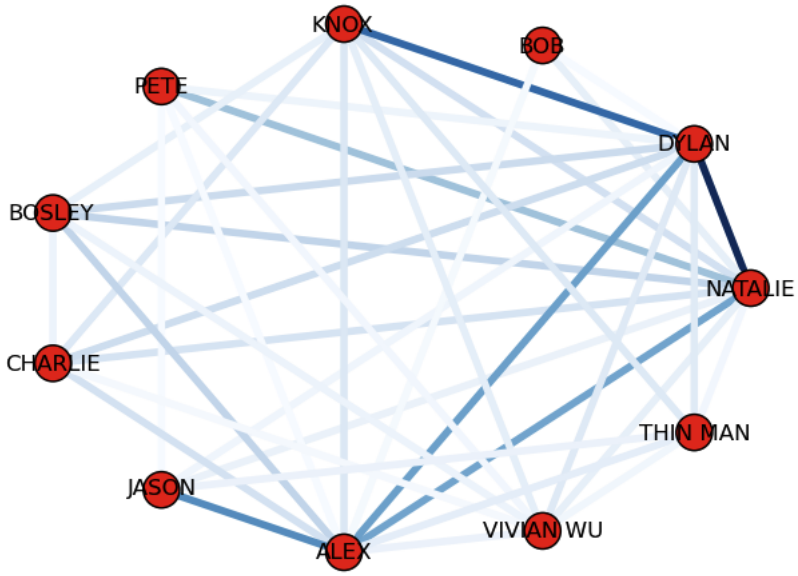
On the other hand, our results suggest that movies tend to have several important main characters who all interact with one another as well as a number of significantly less important minor characters. A good example of this pattern is the movie *Charlie's Angels*, with interaction network shown in Figure 2. There are a few central characters (the angels) forming a strong interaction triangle as well as many more minor characters.

It was particularly interesting to consider the movies and plays which were misclassified by our classifier. One example is the movie *The Nines*. This film is separated into three parts, each one focusing on a different character. The lives of the three main characters intersect in subtle and mysterious ways. Our classifier predicted that this movie, which has many supporting characters involved in several plotlines, was a play. As another example, the classifier incorrectly predicted that Shakespeare's *Much Ado About Nothing*—which focuses on two relationships and a single group of entangled characters, much like a typical romantic comedy film—was a movie.

4.2.3 Play date classifier

Our classifier for play date of publication (pre-1800 versus post-1800) also exhibited strong performance. This classifier used only three features: percentage of existing edges, unweighted clustering coefficient, and mean top character relationship strength. Interestingly, these three properties were all designed for

Figure 2: Interaction network for *Charlie's Angels*



the same goal: to quantify whether the characters were all interacting with one another in the same group (and thus the same plotline) or whether they formed several separate groups. The signs of the feature weights indicate that all three of these properties typically have higher values in newer plays than in older plays, which means that older plays typically have more disjoint groups of characters and more distinct plotlines than newer ones. We gained further insight by examining which examples were misclassified. For instance, Shakespeare's *The Tempest*, an old play that was misclassified as new, is set on an island where most of the characters interact with one another. George Bernard Shaw's work *Man and Superman*, a new play that was misclassified as old, tells the largely disjoint stories of Ann Whitefield and John Tanner. Overall, then, it seems that our classifier was indeed distinguishing works based on whether they had one or many separate character groups and storylines.

Interestingly, the differences between older and newer plays seemed somewhat similar to the differences we observed previously between plays and movies; older plays have more disjoint plots than newer plays just as plays in general have more disjoint plots than movies. Accordingly, we expected to see a big difference between older plays and movies and a smaller difference between newer plays and movies. To test this hypothesis, we built two additional classifiers: one for movies versus older plays (580 and 66 examples, respectively) and one for movies versus newer plays (580 and 107 examples, respectively). To our surprise, we found that the classifier for older plays versus movies achieved an AUC of 0.688 whereas the classifier for newer plays versus movies achieved an impressively high AUC of 0.957. This result clearly contradicted our expectation that movies would be difficult to distinguish from new plays and demonstrated that our initial reasoning, which attempted to place the works on a one-dimensional spectrum based on the number of distinct plotlines, was an oversimplification. We found that the main features which allowed us to distinguish so successfully between movies and new plays were the top character weight variance (higher for movies) and the top character relationship strength variance (higher for new plays). Thus, movies tend to have a greater spread in character importance than new plays, and new plays typically have more pairs of non-interacting main characters.

4.2.4 Movie date classifier

We found that unlike our play date classifiers, our classifiers for movie release dates performed poorly. One issue was that we had relatively few scripts of old movies: only 51 from before 1980 and 25 from before 1960. We initially chose 2000 as a cutoff year for new versus old so that our two classes would have

similar numbers of examples. Our results were also poor when we chose 1990 and 1980 as cutoff years. We have two hypotheses for the poor performance of our classifiers in this task. First, it is possible that we simply did not have enough examples of old movies and that our classifiers would be successful if we obtained more old movie scripts. Second, perhaps old and new movies simply do not differ significantly in terms of their interaction networks. Unlike the play, a literary form which has existed for thousands of years and has therefore had time to change and evolve, movies have existed for less than a century. Accordingly, it is very plausible that old and new movies simply cannot be distinguished well based on their interaction networks.

4.2.5 MPAA, audience, and critic rating classifiers

For the two MPAA rating tasks as well as the audience and critic rating tasks, our classifiers performed poorly. These results made intuitive sense to us as there does not seem to be a strong link between the plot structure of a movie and its rating or reception. In other words, there is a great diversity in the types of movies (and their interaction networks) that are enjoyed by audiences, praised by critics, or given a certain MPAA rating.

4.2.6 Genre classifiers

Our classifiers involving the horror genre all performed rather well. We discovered that horror movies were often characterized by high average top character relationship strength. As mentioned above, this result implies that horror movies are more likely than other genres to contain one simple storyline, which corroborates common stereotypes associated with the genre. This explains why more nuanced horror movies with multiple storylines, such as the *Final Destination* series, were misclassified as non-horror. Interestingly, for some movies whose genre assignment we disagreed with, our horror classifiers often disagreed through misclassification as well. For example, *Aliens*, which was listed by IMSDb as a horror movie, was misclassified as non-horror, which agrees with our assessment that *Aliens* is an excellent sci-fi action movie, unlike its terrifying prequel *Alien*.

Most of our classifiers involving the action genre also performed quite well. We found that action movies are characterized by higher average weighted clustering coefficients than other genres. As action movies frequently consist of two or more factions interacting amongst themselves or with each other in strategizing or fighting, this result makes intuitive sense. It is interesting to note that in one case (action versus comedy), the average weighted clustering coefficient was in fact the only feature selected via the cross-validation step.

While there were certainly many genre classifiers that performed well, we found that some movie genres were far too similar to be distinguished by our classifiers. One notable example is the classifier that differentiates between romance and comedy movies. One issue was that our sample size was limited due to the large number of romantic comedy movies in existence (which were excluded from our data set for this task since they belong to both genres). Furthermore, character interaction networks for unfunny romances and unromantic comedies are already quite similar; comedies such as *Harold and Kumar* often feature a dynamic duo who interact with each other and the world much as main love interests in romance movies do. For other pairs of similar genres (such as crime and action), our classifiers also performed poorly, but the failure of the classifier for romance and comedy movies was perhaps the most dramatic.

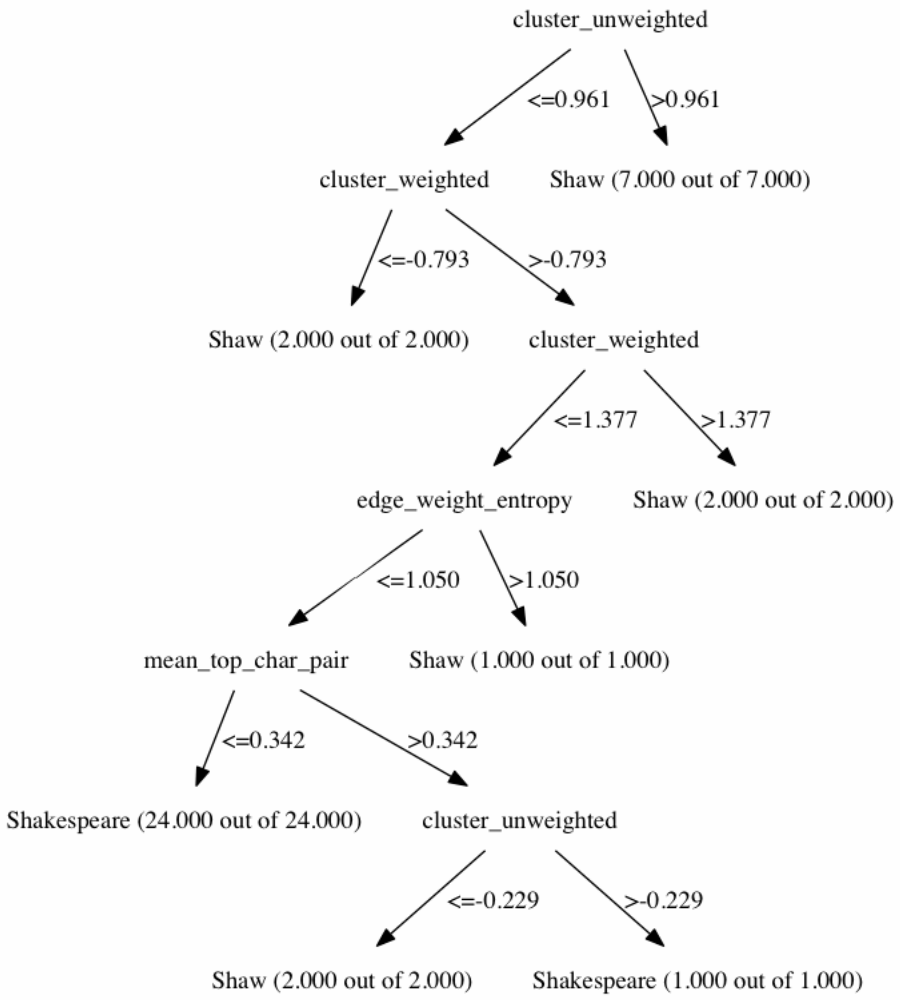
Overall, our analysis of the genre classifiers confirms several common assumptions about what distinguishes certain genres and about which pairs of genres are most similar.

4.2.7 Play author classifiers

Our three classifiers for play author achieved very high AUCs, but we must view these numbers with caution, given the relatively small number of examples. Using just three features, the Shakespeare versus Shaw classifier achieved a perfect AUC of 1.0 on the test set and an AUC of 0.964 on the combined

training and testing set. An examination of the feature weights reveals that Shakespeare plays compared to Shaw plays tend to have lower average clustering coefficients (both weighted and unweighted), as well as lower variance of top character weights. Interestingly, the Shakespeare versus Shaw decision tree (shown in Figure 3), which achieved an AUC of 0.917, also mainly made decisions based on the values of the weighted and unweighted average clustering coefficients, so these features are clearly important in differentiating between the two playwrights. The Shakespeare versus Galsworthy classifier, which also performed quite well, relied primarily on the variance of top character weights (lower in Shakespeare) and number of characters (higher in Shakespeare). It appears, therefore, that one of the defining characteristics of Shakespeare's interaction networks is a large spread in the importance of the main characters.

Figure 3: Decision tree for Shakespeare versus Shaw



5 Future Work

Because this project is the first step into an exciting and largely unexplored area of research, there are very many avenues for interesting extensions to our work. For one, our database of play and movie scripts is far from exhaustive; access to a larger collection of works would strengthen the credibility of our existing results, allow us to attempt further classification tasks, and elucidate new defining characteristics of the studied works. Classification by author is one example of a task for which additional data would

help us greatly. Because there were few authors for whom we had large numbers of works, we were not able to classify by author with tremendous amounts of confidence. If we had more plays and movies with labeled authors, we could more accurately test whether our character interaction networks indeed allow us to distinguish between different authors.

There is also an opportunity to expand into other media including novels and television shows, along with various other types of literature. We would have to apply advanced techniques from natural language processing to meet the challenges associated with identifying interactions in free-form text. Although this would be difficult, it would be very valuable since it would allow us to compare interaction networks across media types other than plays and movies, shedding light on the fundamental similarities and differences between them.

Another direction for future work involves comparing our character interaction networks with real-world networks to reveal whether and how the plays and movies resemble real life. One challenge would be finding real-world networks with weighted edges comparable in size to our play and movie interaction networks. We believe that collaboration networks in academia or networks of workplace interactions have the potential to make good reference networks for such a study.

6 Conclusion

In our project, we successfully extracted character interaction networks from movies and scripts using a novel extraction algorithm, computed informative properties from these networks, and constructed classifiers to distinguish between a multitude of characteristics that define movies and plays. Our classifiers successfully distinguished between media types, publication dates for plays, and different movie genre pairs. An analysis of our results sheds light on the defining properties of the interaction networks of various subsets of our works. With this project, we automated a literary scholar's general approach to extracting meaning from movies and plays, leading us to valuable insights about large numbers of works.

References

[1] J.-B. Michel, et al. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science* **331**, 176, 2011.

[2] F. Mosteller and D. Wallace. *Inference and Disputed Authorship: The Federalists.* Addison-Wesley, 1964.

[3] D. Elson, N. Dames, and K. McKeown. Extracting Social Networks from Literary Fiction. In *Proc. 48th Annual Meeting for the Association for Computational Linguistics*, 138-147, 2010.

[4] F. Moretti. Network Theory, Plot Analysis. *New Left Review* **68**, 2011.

[5] J. Rydberg-Cox. Social Networks and the Language of Greek Tragedy. *Journal of the Chicago Colloquium on Digital Humanities and Computer Science* **1**, 2011.

[6] J. Stiller and M. Hudson. Weak Links and Scene Cliques within the Small World of Shakespeare. *Journal of Cultural and Evolutionary Psychology* **3**, 2005.

[7] R. Alberich, J. Miro-Julia, and F. Rossello. Marvel Universe looks almost like a real social network. *e-print arXiv:cond-mat/0202174*, 2002

[8] C.-Y. Weng, W.-T. Chu, and J.-L. Wu. RoleNet: Movie Analysis from the Perspective of Social Networks. *IEEE Transactions on Multimedia* **11**, 2009.

[9] The Internet Movie Script Database. *IMSDb* 2011. ⟨http://www.imsdb.com/⟩.

[10] Project Gutenberg. *Project Gutenberg* 2011. ⟨http://www.gutenberg.org/⟩.

[11] The Complete Works of Shakespeare. *MIT* 2011. ⟨http://shakespeare.mit.edu/⟩.

[12] EOneill.com EText Archive. *EOneill* 1999. ⟨http://www.eoneill.com/texts/index.htm⟩.

[13] Read Plays Online - Read Print. *Read Print Library* 2011. ⟨http://www.readprint.com/⟩.

[14] The EServer Drama Collection. *EServer* 2011. ⟨http://drama.eserver.org/plays/⟩.

[15] Rotten Tomatoes. *Flixster, Inc.* 2011. ⟨http://www.rottentomatoes.com/⟩.

[16] J.-P. Onnela, et al. Intensity and coherence of motifs in weighted complex networks. *Physical Review E* **71**, 2005.

[17] M. Robnik-Sikonja and I. Kononenko. An adaptation of relief for attribute estimation in regression. In *Proc. 14th ICML*, 296-304, 1997.