Computational Plot Planning: A Temporal Social Network Approach

Haoran Ding
Media Technology MSc Programme, Leiden University
bitdhr@hotmail.com

Thesis supervisors:
Frank Takes, Leiden University
Pádraig Mac Carron, University of Oxford
Maarten Lamers, Leiden University

Abstract

Computational plot planning is an important topic and challenge in the field of artificial intelligence and computational creativity. Current methods dealing with the problem of story plot planning face difficulties in supporting long storylines and complex social structures. To solve this, we propose a new approach: using a temporal social network as a reference to plan future social interactions among characters in the developing plot. In this study, we analyze the development of social networks in stories through time and make predictions on future social network structures. Furthermore, we address three special properties of story social networks that each bring unique problems with them: newly introduced characters at each chapter, characters leaving the story, and special narrative focus. We used a LSTM recurrent neural networks model to analyze four story datasets: the Odyssey, Iliad, Lord of the Rings and Game of Thrones. Based on this, our model attempts to predict a social network structure that fits the structure of the existing story. The model successfully predicted story social network structures that were similar to those in the original stories and shows promising potential for the plot planning field.

Keywords: Plot planning, social network analysis, temporal networks, recurrent neural networks.

1 Introduction

Narratives, and storytellings in particular, are one of the most important parts of human social lives [10, 33]. Since the 1970s, storytelling has been a topic of interest in Artificial Intelligence: computational plot planning has been researched extensively by both computer scientists and linguists in the field of computational creativity and intelligent systems [4, 25].

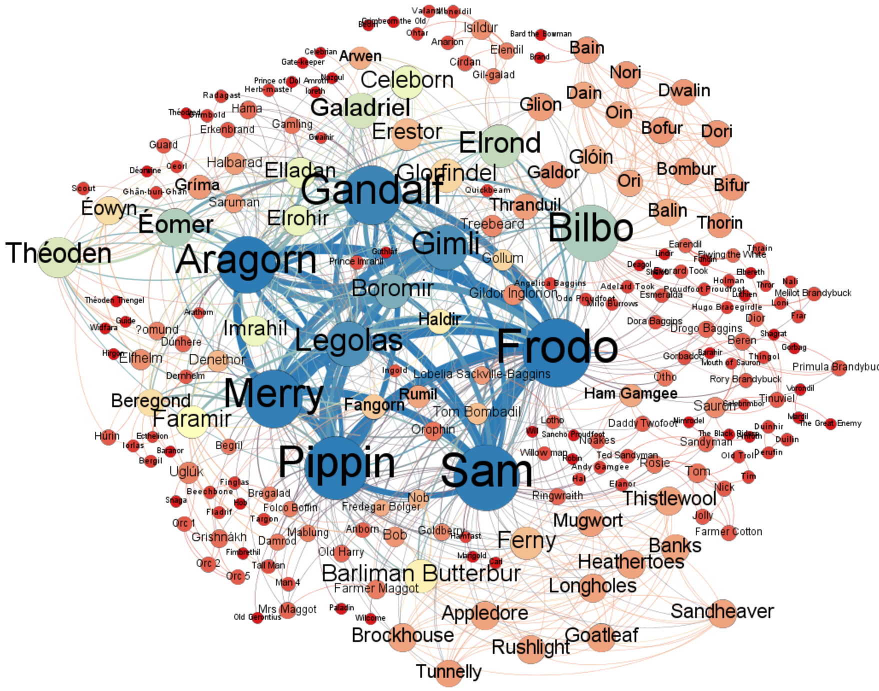

Figure 1: The social network visualization of the Lord of the Rings dataset. In this graph, the more edges one node has connected to it, the larger its size is and the closer to the color blue. The more frequently the interactions happened between the nodes, the thicker the edge is.

Computational plot planning is a study where computational models are expected to generate logical or even creative narrative plots. Plot planning methods introduced in the past predominantly focused on sentence grammar, story schemas and content-based autonomous agents [24]. Current story generators focus mainly on generating the content of the story, as well as trying to construct proper sentences [23]. However, it is difficult for these current systems to support long storylines and social structures within these stories, especially when many characters are involved.

In stories, interactions between characters are one of the most essential parts of the storytelling. As a story progresses, the social structure within the story evolves. In this thesis, these social interactions were described by two models: the social network model and the temporal social network model. The social network model treats characters in a story as nodes and the interactions among these characters are represented by edges. By considering all character interactions in a story, we can extract a social network model of the whole story and analyze the narrative social structures [21]. In the temporal social network model, a specific chapter or scene number is used as the time frame mark. This way, a temporal network can be extracted from each chapter, representing the social network at that point in the story [31]. In terms of plot planning, knowing what a future temporal network looks like can make it easier to plan interactions between characters and thus help in developing the plot. An example social network structure visualization, based on the Lord of the Rings books, is shown in Figure 1.

In this paper, our goal is to analyze the development of social networks in stories through time and make predictions about the social network structure in the following chapter/scene. To achieve this, we will build a computational model for network prediction, which will attempt to predict the network structure in the upcoming story based on temporal social network data. This predicted network structure represents future interactions between characters. Overall, the story social network will provide a global understanding of a story, as well as a social relationships for the plot. This bridges two fields in computer science: computational plot planning and temporal social network analysis, using the social network structures to provide a reference for plot planning.

This paper is organized as follows. First, we discuss preliminaries and related work for this study (Section 2). This is followed by a formal description of our research question (Section 3). We describe the method in Section 4. In the sections that follow, we describe the datasets that were used (Section 5), the experimental setup and results (Section 6), followed by a conclusion and discussion (Section 7). Finally, we discuss possible future directions of this research (Section 8).

2 Preliminaries and related work

In this section, we introduce the concepts and properties of a social network, as well as the link prediction problem and the LSTM recurrent neural network used to perform the prediction.

2.1 Social network analysis

In the following sections, we introduce the network concepts and properties as well as the static network and temporal network used in this thesis.

2.1.1 Network concepts and properties

An undirected network (graph) $G = (V, E)$ is a set of nodes (vertices) V and a set of links (edges) E. The number of nodes N in a network is given by $|V|$ and the number of links L is given by $|E|$. A link between node u and node v is denoted as (u, v). A link can also have numeric attributes. In our study, each link has two attributes: $weight(u, v)$ and $cost(u, v)$.

In our social network, the nodes represent the characters in the story and the links indicate the interactions among characters. The $weigh$t of a link represents the number of times two linked characters had an interaction so-far in the story. The higher the $weight$, the more frequently they interacted. Based on the *weight* concept, we use

$$cost(u, v) = 1/(\log(weight(u, v)) + 1)$$

to model the cost between node u and v. In this study, the minimal *weight* of a link is 1, so the range of *cost* is $(0, 1]$. The *cost* indicates the cost of information flowing through the social network [5, 11, 33]. The higher the *weight* between two nodes, the lower the *cost* and the more efficient the interaction will be. All the network datasets used in this study are undirected, as all the interactions we considered are bilateral ones.

Three network properties are used to characterize the network: average clustering coefficient, average node-to-node distance and average degree. The clustering coefficient measures the tendency of nodes in a graph to cluster together. The average clustering coefficient of a social network is significantly higher than that of a random graph [21, 35]. The average node-to-node distance (average path length) is computed using the largest connected component of the network. It measures the connectivity of the network and tends to be very short in social networks [16]. The average degree indicates the average number of people one character had interacted with in total.

2.1.2 Static network and temporal network

As mentioned earlier, we generate a temporal network G_t at each chapter t of the story. The range of integer t is between 1 and the last chapter of the prediction T. The temporal network G_t represents the interactions and social structure in chapter t. We can also create a static network by accumulating the previous temporal networks until a specific time frame. A static network Gs_t represents all the past interaction history of the story up until chapter t. For example, the static network Gs_10 at chapter 10 of the story contains all characters and interactions mentioned from chapter 1 to chapter 10.

2.2 Link prediction

The link prediction problem is about predicting the absence or presence of a future link between all pairs of nodes within a graph [27]. There are two types of link prediction [27]: structural and temporal. Structural model, which means that given a partially observed graph, we are expected to predict the edges of the unobserved graph; temporal model, where we have a sequence of fully observed graphs and our goal is to predict the graph of the next time frame. We performed the temporal link prediction in this study, as our goal is to predict the future structure of the story network. In our link prediction problem, given temporal networks $G_1 = (V_1, E_1)$ to $G_t = (V_t, E_t)$, the model is expected to predict the links of the network $G_{t+1} = (V_{t+1}, E_{t+1})$. The number of previous chapter taken into consideration is the recurrent window size, which will be discussed in Section 2.5.

Existing link prediction models can be classified as either unsupervised or supervised [27]. Unsupervised models compute scores for each pair of nodes based on the topological information of a graph. The model then uses these scores as a prediction measure to determine

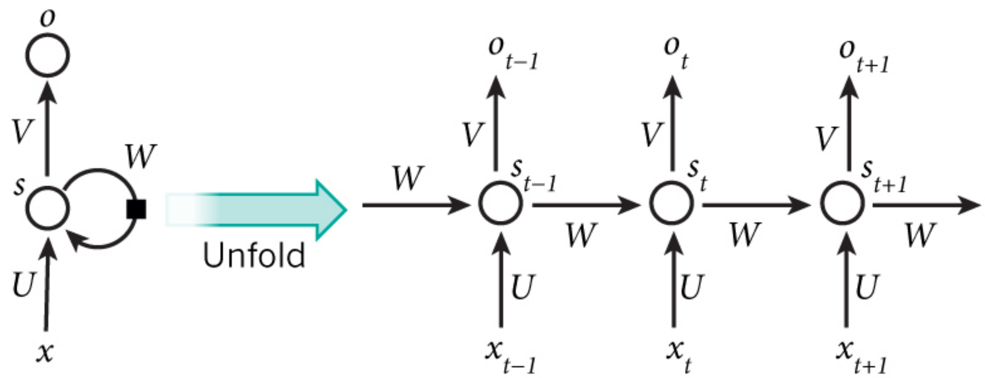

Figure 2: Classical representation of a recurrent neural network. [3]

whether there is a link between a pair of nodes or not [27]. The famous scores are the Adamic-Adar [1] and Katz score [20]. The scores of unsupervised models are predefined and do not involve any learning process.

There are multiple options to choose from when using a supervised method: graph regularization model [32], latent class models [2,9], latent feature models [26,27], and the method we applied in this paper: feature-based models [18, 34]. The feature-based model takes a graph's topological features (like degree or common neighbors) and side information (like temporal information) as the training features for supervised machine learning algorithms such as Recurrent Neural Networks. We chose to use the feature-based model as it is able to combine both the topological and temporal information [27]. Furthermore, it is flexible towards the data imbalanced problem and it supports different graph scales [27]. Moreover, the feature-based model is adaptive to challenging situations, such as introducing new characters in a story for which no prior social relationships data exists. For characters that are introduced later (new nodes), we can put zero degree nodes in the training dataset. The feature-based model will recognize patterns in the topological features of existing nodes, from them which ones will have the higher possibility of getting attached by the new nodes.

2.3 LSTM Recurrent neural network

Neural network models are known for their performance in pattern recognition, computer vision, speech recognition, natural language processing, etc. Traditional Neural Network models often do not take the history of the input data into account, whereas the RNN model performs better when there is sequential input data involved because it takes the previous hidden layer as an extra input (Figure 2) [3]. In this study, the history of a story's social network structure would affect future network structures. Therefore, we chose to use the recurrent neural network model.

The recurrent model performs the same task for each input element in a sequence, with the outputs being dependent on the previous computation. However, the main problem of RNNs is that during training, components of the gradient vector can grow or decay exponentially over long sequences, resulting in a vanishing or exploding gradient [13]. This problem makes it difficult to train the RNN model to retain long sequence information. To solve the issue of learning long-term dependencies, we used the Long Short-term Memory Network (LSTM) model proposed in 1997 [13]. These *gated* RNNs are based on the idea of creating paths through time that have derivatives which neither vanish nor explode [3]. An important addition has been to make the weight on the recurrent layer's self-loop not fixed, but conditioned on the context [12].

The link prediction is a binary classification problem. The output can be either class0: there is no link between this pair of nodes in the next time frame, or class1: there is a link. We chose the LSTM model as it takes the history of the input data into account. Furthermore, the LSTM can also handle long sequences in data, which is necessary as stories can include shifts between main or side storylines, or mention important relationships early on in the story which can then affect future relationships much later. In these two situations, the related interactions between characters were not mentioned in the recent past, so we need a way to consider long sequences in the data. The LSTM model is suitable for such purposes.

2.3.1 Data imbalance and RNN parameters

In network link prediction, we often face the dataset imbalance problem. That is, compared to the number of linked node pairs, the number of non-linked pairs is exceptionally large. The problem for the data imbalance is that many algorithms focus on the classification of the majority sample, while ignoring or misclassifying the minority sample [19]. In our study, linked pairs are very important as we use them to structure the future network, but they are the minority sample. Our approach to this problem is to put more weight on the class1 ('there is a link' cases) to balance the dataset, i.e. both classes would have a similar amount of total weight. The class weight for each dataset is computed based on the ratio of the total number of possible links to the average number of links in each chapter.

The window size of the recurrent layer indicates how many previous temporal networks we take into consideration. That is, if we set 10 as the window size, the model takes the training features of the 10 previous chapters as input for each prediction. The window size differs among datasets and is determined according to the total number of chapters of the story.

3 Problem statement

In this thesis, we focused on social interactions among characters within the story and have the model support the social structure of the plot. We divided the story plot planning problem into two sub-questions:

1. Can we predict a story's future social network structure, based on a story's previous social network data?

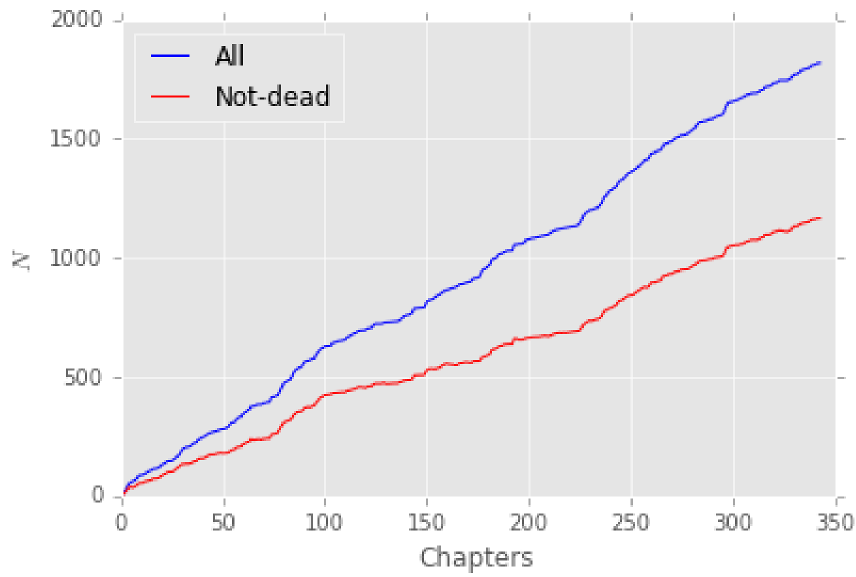

Figure 3: Network size N growing over chapters, Game of Thrones

2. Can we translate the predicted network back to narrative plots?

In this thesis, we are mainly focusing on answering the first question. We treat the first question as a network link prediction problem: based on the historical structure of the social network of a story at given time, we would like to predict which links will form between characters in the temporal network at the next time frame. Then, we compare the network's predicted links to the actual network (target network) and measure the quality of the prediction.

Story social networks set themselves apart from other social networks in terms of special properties, such as the introduction of new characters, characters leaving the story line for various reasons such as death, and the narrative focus shifting between main story line and side story lines. Figure 3 shows the growth of the network size N over the developing of the Game of Thrones story, where new characters are introduced in each chapter of the story. The difference between the red line and blue line represents dead characters. Unlike most social interaction-formed temporal networks (e.g., email contact networks, social media networks) which document all interactions between individuals, story social networks only show small, certain parts of them at each chapter/scene. We see the story's fictional world through a camera lens. For example, in the Lord of the Rings, a whole chapter could only include interactions between Frodo and Sam despite the existence of other characters. These properties make story social networks more difficult to predict. We will discuss in Section 4.2 how we deal with these problems.

Static network				Temporal
degree(u)	weighted degree(u)	clustering coef(u)	pagerank(u)	degree(u)
degree(v)	weighted degree(v)	clustering coef(v)	pagerank(v)	degree(v)
shortest path (u,v)	common neighbors(u,v)	jaccard coef(u,v)		has path(u,v)

Table 1: Training features extracted from static and temporal networks

4 Methodology

In this section, we first describe the data used to train the model as well as the evaluation metrics for the model. Followed by the discussion of model structure, finally our approaches to deal with the three special properties of story social network.

4.1 Training data

We discuss the topological features of networks we use for prediction and what these features mean in the context of stories. Furthermore, we give an overview of the training data extracted from both static and temporal networks (Table 1).

4.1.1 Topological features

Based on the research of van Engelen et al. [34], the topological features we used for link prediction can be divided into three classes: node features, neighborhood features and path features. These three types of features indicate the structure of a social network from different perspectives. The definition of these features are as follows.

Node features:

- Degree: the node degree is the number of edges a node has connected to it. For our dataset, it means the number of characters that one character has interacted with.
- Weighted degree: total weight of all edges of the node. In terms of story networks, the weighted degree is the total number of interactions a character has had so far.

Neighborhood features:

For convenience, we define $\Gamma(v) = \{u \in V : (u, v) \in E\}$ as the neighborhood of node v in V. In terms of our story networks, the neighborhood of a node is the set of people that a character has interacted with so far. Usually, this represents a character's circle of friends.

- Common neighbors: the number of neighbors two nodes have in common, $|\Gamma(v) \cap \Gamma(u)|$. It is the number of mutual friends for two story characters.
- Jaccard coefficient: the number of common neighbors two nodes have relative to the total number of the distinct neighbors they have, $|\Gamma(v) \cap \Gamma(u)|/|\Gamma(v) \cup \Gamma(u)|$. The Jaccard coefficient measures the ratio of mutual friends against the set of total friends of two characters.

- Local clustering coefficient: it measures how close to a clique the neighbors of a node are. $2T(u)/(deg(u)(deg(u)-1))$ where $T(u)$ is the number of triangles through node u and $deg(u)$ is the degree of u.

Path features:

- PageRank: The PageRank algorithm computes the global importance of a node [29]. PageRank indicates the global influence of a character within the story's social network.

- Shortest path: the length of the shortest path between two chosen nodes. This closeness measure is often referred to as distance. It is computed with the cost attribute of links in Section 2.1.1 and outputs the total cost of the shortest path between two nodes.

4.1.2 Training dataset and pre-processing

The choice of training features is based on previous work by Engelen et al. [34]. We also added the PageRank of each node to measure the global influence of that node in the static network. Because both the static social structure and recent interactions can play a role in predicting the future interactions, we included both static and temporal features in our prediction model. There are a total of 14 features for each possible link (u, v) (Table 1). Eleven features are generated from the static network and three from the temporal network (the 'temporal' column). The temporal feature $haspath(u, v)$ denotes in binary value of whether or not there is a path between node u and v in the temporal network. Overall, the training data can be represented as a three dimensional matrix (possible links, chapter, features). The output matrix is two dimensional (possible links, chapter). To keep the features on the same scale throughout the story time line, each feature will be normalized (L^2 norm) within each chapter before being fit into the model.

4.2 Model evaluation metrics

To evaluate the binary classification performance on an imbalanced dataset, we use the Precision-recall and F1 score metrics [30]. In addition, we use the degree distribution and IM distance to measure the difference in structure between the predicted network and the target network [15, 21].

4.2.1 Precision-recall and F1 score

Precision-recall is a useful measure for the prediction quality when the dataset is imbalanced. In binary classification (not-linked class0: negative, linked class1: positive), we define:

- True Negative: case was negative and predicted negative

- True Positive: case was positive and predicted positive

- False Negative: case was positive but predicted negative

- False Positive: case was negative but predicted positive

The precision, recall and F1 score are defined as:

- $Precision$ = True Positive/(True Positive + False Positive)
- $Recall$ = True Positive/(True Positive + False Negative)
- $F1 = 2Precision*Recall/(Precision + Recall)$

A good prediction result is expected to have a F1 score close to one.

4.2.2 Degree distribution and IM distance

As we included link prediction cases involving new nodes, precision-recall is not always a good measure for the model. For example: if at chapter t we have node a and the target link to predict is between node a and the new node b in time frame $t + 1$, a model output that instead predicts there is a link between node a and a new node c will receive a low F1 score. However, we would still consider this a good prediction: in plot planning, when we introduce a new character, the name of the character does not matter as much as the interaction. We can solve this problem by looking into the structure of the two networks, because if we do not consider the identities of nodes, the predicted and target networks have the same structure. Therefore, we propose to use two measures for the network structure: KS degree distribution and IM distance. These can help us evaluate whether the prediction is able to keep a similar network structure as the target network throughout the temporal development.

The degree distribution of a network is the probability distribution of node degrees over the whole network. The degree distribution measure can be used to distinguish different types of networks [21, 28]. Degree distributions of social networks often follow a power law [28]. By measuring whether the predicted network and the target (actual) network have the same degree distribution, we could observe if the model is able to capture the structure of the network. We performed the KS measure on the networks to determine network differences on degree distribution [6–8]. The null hypothesis is that the two networks are drawn from the same degree distribution [36]. In this case, p value gives the probability that the two observed network distribution are similar [36]. If the p value is small ($p < 0.05$), we could conclude that two networks are likely to have come from different degree distribution. In this thesis, we use the low p value ($p < 0.05$) as indicating of a bad performance of the model as it may indicate a "mismatch" between the two distributions.

In addition, we also compute the IM network distance [15] between the predicted and target network. It is a very robust measure where we make quantitative comparisons between networks [14]. The IM distance quantifies the structure differences between two networks and it is also applicable when two networks have different sizes. The IM distance will be close to 1 when two networks are maximally different (0 means identical, 1 means opposite). A good predicted network will have similar degree distribution as the target network and a small IM distance.

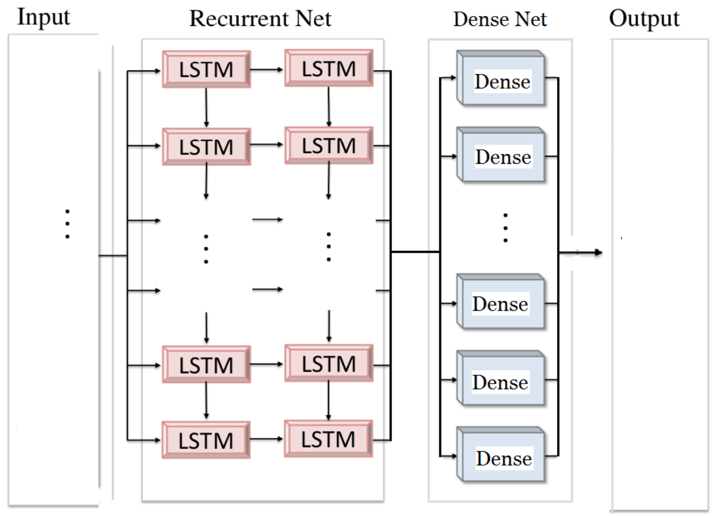

Figure 4: LSTM model structure

4.3 Model structure

In this thesis, we are aiming to construct a generic model to predict the future statue of story social networks. We applied a multi-layer LSTM feature-based link prediction model. The choice to use the LSTM neural network is motivated by the need to be able to consider previous data, as previous temporal network structures could affect future network structure. We trained the LSTM neural network using training manually extracted from several books in previous studies [21, 22]. If a face-to-face interaction occurred in the book, the link between these two characters is documented.

Figure 4 shows the structure of our model. The training data was first fit into two layers of LSTM units, followed by one fully connected layer (dense layer) which then outputs a binary number indicating whether there will be a link in the next time frame or not. The structure was constructed based on previous research [3, 12]. We first started with a single layer LSTM recurrent net and then added in another layer to improve the model performance of precision-recall, KS degree distribution and IM distance measures. The parameters of the model were first chosen empirically and then optimized manually. For each dataset, we chose the window size based on the number of total chapters in a book. The class weight we use is based on the ratio of average link numbers at each chapter against the total possible links. Depending on the performance of the model, we optimized the parameter. RMSprop [17] was chosen as the gradient descent optimization algorithm due to its high performance for recurrent neural network.

4.4 Problems in story social networks

Three problems were mentioned at the end of Section 3 regarding the special properties of story social networks: newly introduced characters, character leaving the story, and special narrative focus. The approaches we took to solve these problems are as follows:

- Introduction of new characters: in the training matrices we use to represent the temporal network, we added zero degree nodes. The model is trained to use the same features of link prediction to pick up the new node's preferential attachment patterns. We left the situation in which links formed between two new nodes for future study, as in the case there is no topological feature from the existing network can be used for prediction such links.

- Characters leaving the story: among the four datasets we used in this thesis, the Game of Thrones and the Lord of the Rings datasets documented the death of characters. While generating social networks based on these two datasets, we treated dead characters as follows: once a character died, their nodes as well as their edges were removed from the static network. The training features related to the dead character were set to zero in all chapters following their death. Unfortunately, in terms of predicting the possibility of death of a character based on the topological features, we found the pattern is rather random. Thus, it is difficult to train a computational model to predict the death. This problem may be of interest in future studies.

- Special narrative focus: story social networks can focus on a few specific interactions in each chapter or scene, which makes the data even more imbalanced. For example, a chapter in a Lord of the Rings book may be dedicated to a interaction between Frodo and Sam. Furthermore, if a story involves multiple storylines and shifts between them, it also makes the social network harder to predict. We used two approaches to solve this issue: first, the class weight for the prediction model varies per story. This customizes the prediction model, to a certain extent, to the corresponding narrative style. Second, instead of predicting the social structure for only one future chapter, we set the model to predict links for the next two chapters. This could reduce inconsistency caused by shifts in the storyline's narrative focus and it may also help to balance the data.

5 Datasets

Table 2 provides an overview of the social networks datasets used in this study, including the source and network properties. Two of them, Iliad and the Odyssey, are major ancient Greek epic poems attributed to Homer. We also included two popular contemporary epic fantasy novels: Lord of the Rings and Game of Thrones (A Song of Ice and Fire). The Game of Thrones dataset is extracted from the five current books, with 'A Dance with Dragons' as the last book in the dataset.

For each network, the number of nodes, number of links, average number of newly introduced characters and dead characters at each chapter and average clustering coefficient (CC) are listed in table 2. Average node-to-node distance of the largest connected component is also listed. We excluded zero degree nodes: that is, monologue and mentalization of a character were not taken into account in this study. We also excluded hostile links (overall less than 5%) to reduce the imbalanced data problem. At each chapter, Iliad has the highest

Dataset	Chapters	Nodes	Links	Avg.new	Avg.dead	CC	Avg.dist	Avg.deg
Odyssey	24	300	995	11.652	NA	0.445	3.388	6.633
Iliad	24	640	2330	26.217	NA	0.435	3.784	7.281
Lord of the Rings	63	216	850	3.032	0.159	0.612	2.947	7.870
Game of Thrones	343	1818	14504	5.301	1.901	0.644	3.415	15.927

Table 2: Dataset description. For each network: chapter numbers, the number of nodes, number of links, average number of newly introduced characters (Avg.new) and dead characters (Avg.dead) at each chapter, average clustering coefficient (CC), average node-to-node distance (Avg.dist) as well as average degree (Avg.deg) are listed

number of newly-introduced characters compared to the other three. The death information for The Odyssey and Iliad is not available. Game of Thrones has a higher character death rate in comparison with Lord of the Rings.

6 Experiments and results

In this section, we will first discuss the model parameter choices for the experiments, followed by the results and a discussion.

6.1 Experimental setups

Based on the total number of story chapters, we trained the model with two thirds of the data. We then used the trained model to generate predictive networks, using the first part of the dataset as input. Table 3 shows the experimental parameters for each dataset. The 'Chapters' column shows how many chapters the dataset has and the 'Predicted chapters' column indicates how many chapters were predicted after the training. The 'Window size' is the number of previous chapters our model received as input data to predict the next one. Finally, the 'Class weight' column represents the weight we assigned to class1: linked pairs meanwhile the class weight of class0 is 1.

During the prediction process, each newly generated network Gp_t was treated as the input data when generating the following networks Gp_{t+1}. This way, with an initial dataset, the model is able to continue predicting more chapters. However, one of the disadvantages of the continuous prediction is that the recurrent network model output will mostly be narrowed down to interactions within the last few chapters (depending on the chosen window size) of the given initial dataset. This narrow focus on specific interactions will be reinforced as prediction continues, which will lead to a significant difference between the target and the predicted networks. This will be especially obvious if the model is expected to predict a large number of chapters and all their associated networks. In terms of plot planning, this will cause the storyline stay within a certain scope of interactions and make it difficult for the storyline to develop.

Dataset	Chapters	Predicted chapters	Window size	Class weight
Odyssey	24	8	6	9.149
Iliad	24	8	6	10.754
Lord of the Rings	63	20	10	9.905
Game of Thrones	343	72	20	61.064

Table 3: Experimental parameters for each dataset

Dataset	**Degree distribution**	**IM distance**	**Precision**	**Recall**	**F1**
Odyssey	0.068	0.084	0.842	0.866	0.854
Iliad	0.843	0.030	0.973	0.880	0.924
Lord of the Rings	0.978	0.038	0.884	0.807	0.844
Game of Thrones	0.00	0.170	0.469	0.376	0.418

Table 4: Experiment results of four datasets. Degree distribution column shows the p_value from the KS measure of the predicted and target networks; IM distance shows the distance measure between the two networks

6.2 Results and discussion

Table 4 shows the performance of our model on the four datasets according to five measures: the precision-recall measure, KS degree distribution measure and IM distance metric. All the measures were computed for the final predicted static network against the target network. For example, for the Odyssey dataset, the model generated the predicted networks for the last 8 chapters and accumulated these in a final predicted static network. This predicted static network was then compared to the target static network that was based on the Odyssey's original last 8 chapters. The results for the degree distributions of the predicted networks and target networks for all four datasets are shown in Figure 5. From the figures, we can see that the degree distributions of three predicted networks are similar to their target networks, with Game of Thrones being the exception. From the degree distribution result for the Game of Thrones dataset, it appears that the actual story network had a broader focus of the interactions than the predicted network (predicted blue line is below the target red line and a low score on the Recall measure in Table 4). This may be because it focused on specific interactions in the input data and continued to re-use only these in the continuous predictions, resulting in a narrowly-focused output. The narrow focus was reinforced by the large number of predicted chapters (72).

The results for the Iliad, Odyssey and Lord of the Rings datasets show high scores on the F1 scores, as well as small IM distances. Also the KS p values are not smaller than 0.05. This means that the model made good predictions for these three datasets on all these measures. The model predicted social network structures that were similar to the structures found in the original works. In addition, the model was able to pick up interaction patterns between characters over the narratives. Furthermore, with the Iliad dataset which has the highest average number of newly-introduced characters per chapter, our model succeeded in predicting links involving new nodes, which means it could introduce new characters and

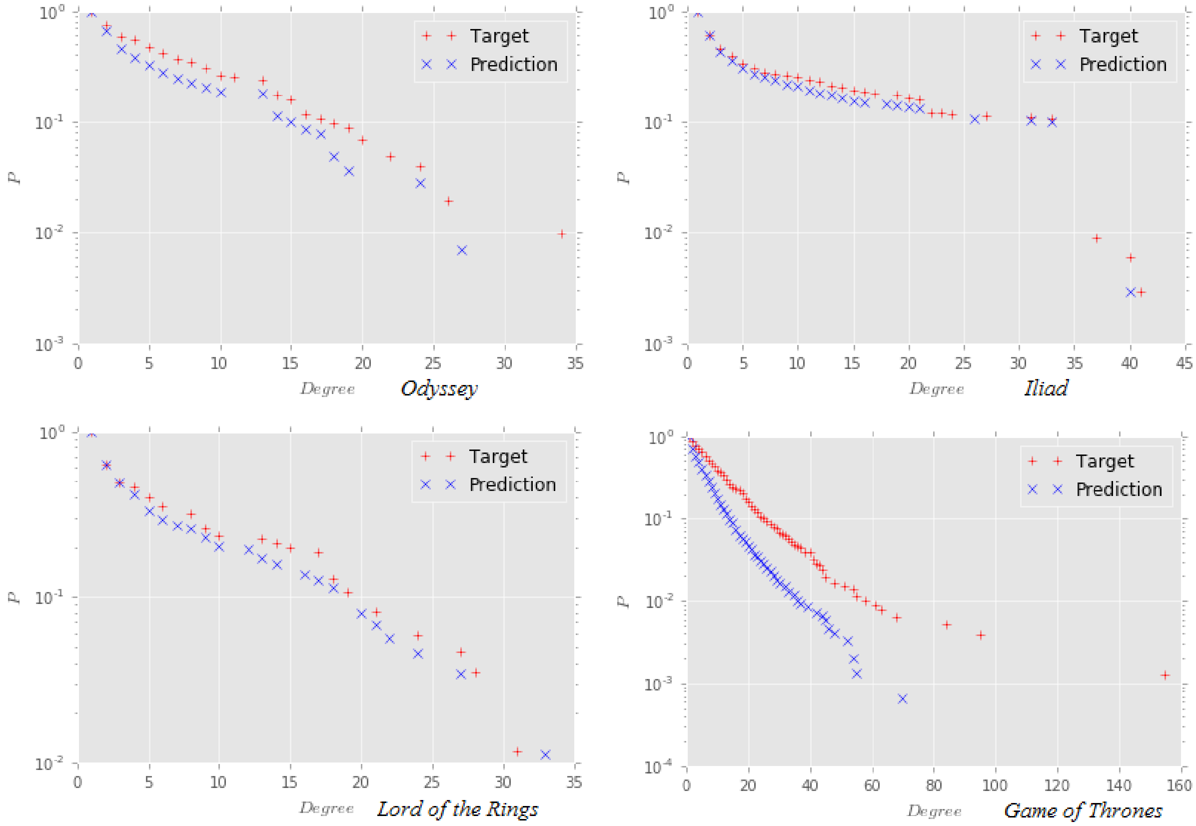

Figure 5: Accumulated degree distributions of the predicted and target networks, from top left to right: Odyssey and Iliad; bottom left to right: Lord of the Rings and Game of Thrones

predict social interactions with them.

However, for the Game of Thrones dataset, the model failed in keeping track of both the interaction patterns and the social network structure. Besides the long sequence of prediction, this may be because the Game of Thrones dataset has a high death rate at each chapter (Figure 3, Table 2), which results in too high a rate of change in network structures for the model to catch. This is especially the case when an important character dies: his/her leave removes many links from the network, resulting in distinct changes in the topological features of his/her neighborhood. More importantly, it may also be that there are too many factions (communities) in the Game of Thrones story social network. This leads to more frequent narrative focus shifts. Even when using a relatively large window size (20), the model still found it difficult to make accurate predictions. In the future, we plan to apply this model to each community in the Game of Thrones dataset, which might result in a better performance.

We now answer our research question: can we predict a story's future social network structure, based on a story's previous social network data? For three datasets, our model was able to properly predict future social networks after being trained with existing data. It provided good performance against the three story social network problems. This means we could use the output from the model as a reference or inspiration for future interactions within a story social network, as well as a backbone for plot planning. The model's prediction will maintain logic in the future plot and support the development of the story social structure.

7 Conclusion and future work

In this study, we brought quantitative analysis into the storytelling field and researched the use of a story's social network as a tool for plot planning. The social network within a story can provide us with information on the social structure and interaction patterns over the narrative. To predict future social networks in stories, we performed link prediction based on four story datasets. Our model made good predictions on the story social structures that closely resembled those of the actual stories in three out of four cases. In the case of Game of Thrones, problems with narrative shifts between communities as well as a high rate of character introduction and death hampered the model's performance. Overall, we are able to accurately model how a social network evolves as a story progresses, but face difficulties in modeling the more "creative" parts of a story such as character death and narrative shifts.

There are two aspects that we would like to address in the future development of our model. Firstly, our study currently only considers non-hostile or friendly interactions among characters. Even though hostile links occur far less often than friendly ones, they are essential for storytelling. Hostile or enemy relationships spark curiosity and interest in the audience. Including hostile links within the social networks can improve the model and make it possible to predict the conflict part of the story social structure. Secondly, as our model attempts to learn patterns based on network features, the predicted plot might become boring as the predictions are based on existing patterns rather than on creating something new. Therefore, in the future, surprise-generating functions such as turning friendships into enmity or the removal of characters will also be integrated into the model. This may make the story more creative and fun to read.

Finally, we would like to give some directions on answering our second research question: translating the social network back to narratives. Our current dataset does not specify what kind of interactions occur between the two characters. If interaction data with specific events or details (e.g., had a lovely conversation, went out to a bar, cooking together, etc.) was documented, we could predict both the link and the possibility of a specific type of event occurring. A plot can be generated based on these events of the social network links. With these developments, we may come closer to models that can generate stories that are as intriguing and creative as many works of fiction today.

References

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(5):1981–2014, 2008.

[3] Yoshua Bengio, Ian J Goodfellow, and Aaron Courville. Deep learning. *Nature*, 521:436–444, 2015.

[4] Selmer Bringsjord and David Ferrucci. *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine*. Psychology Press, 1999.

[5] Herbert H Clark. *Using language*. Cambridge University Press, 1996.

[6] William J Conover. A kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, 67(339):591–596, 1972.

[7] William Jay Conover and William Jay Conover. *Practical nonparametric statistics*. Wiley New York, 1980.

[8] Wayne W Daniel et al. *Applied nonparametric statistics*. Houghton Mifflin, 1978.

[9] Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj. *Generalized blockmodeling*, volume 25. Cambridge University Press, 2005.

[10] Robin Dunbar, Clive Gamble, and John Gowlett. *Thinking big: how the evolution of social life shaped the human mind*. Thames & Hudson, 2014.

[11] Susan H Foster-Cohen. Relevance: communication and cognition. *Second Language Research*, 16(1):77–92, 2000.

[12] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Giuseppe Jurman, Samantha Riccadonna, Roberto Visintainer, and Cesare Furlanello. Biological network comparison via ipsen-mikhailov distance. *arXiv preprint arXiv:1109.0220*, 2011.

[15] Giuseppe Jurman, Roberto Visintainer, Michele Filosi, Samantha Riccadonna, and Cesare Furlanello. The him glocal metric and kernel for network comparison and classification. In *Proceedings of the IEEE International Conference on*, pages 1–10. IEEE, 2015.

[16] Ralph Kenna and Pádraig Mac Carron. Maths meets myths: Network investigations of ancient narratives. In *Proceedings of the Journal of Physics: Conference Series*, volume 681, page 012002. IOP Publishing, 2016.

[17] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Ryan N. Lichtenwalter, Jake T. Lussier, and Nitesh V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July*, pages 243–252, 2010.

[19] Rushi Longadge and Snehalata Dongre. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.

[20] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[21] Pádraig Mac Carron. *A network theoretic approach to comparative mythology*. PhD thesis, Coventry University, Coventry, 2014.

[22] Pádraig Mac Carron and Ralph Kenna. Universal properties of mythological networks. *EPL (Europhysics Letters)*, 99(2):28002, 2012.

[23] Neil McIntyre and Mirella Lapata. Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, pages 217–225. Association for Computational Linguistics, 2009.

[24] Neil Duncan McIntyre. *Learning to tell tales: automatic story generation from corpora.* PhD thesis, University of Edinburgh, 2011.

[25] James R Meehan. Tale-spin, an interactive program that writes stories. In *Proceedings of the IJCAI*, volume 77, pages 91–98, 1977.

[26] Aditya Krishna Menon and Charles Elkan. A log-linear model with latent features for dyadic prediction. In *Proceedings of the IEEE International Conference on Data Mining*, pages 364–373, 2010.

[27] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.

[28] Mej Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351, 2005.

[29] L Page. The pagerank citation ranking : Bringing order to the web. *Stanford Digital Libraries Working Paper*, 9(1):1–14, 1998.

[30] David Martin Powers. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.* Bioinfo Publications, 2011.

[31] Sandra D Prado, Silvio R Dahmen, Ana LC Bazzan, Padraig Mac Carron, and Ralph Kenna. Temporal network analysis of literary texts. *Advances in Complex Systems*, 19(03):1650005, 2016.

[32] Rudy Raymond and Hisashi Kashima. Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 131–147, 2010.

[33] Max van Duijn. *The lazy mindreader: a humanities perspective on mindreading and multiple-order intentionality*. PhD thesis, Leiden University, 2016.

[34] Jesper E van Engelen, Hanjo D Boekhout, and Frank W Takes. Explainable and efficient link prediction in real-world network data. In *Proceedings of the International Symposium on Intelligent Data Analysis*, pages 295–307. Springer, 2016.

[35] Duncan J Watts and Steven H Strogatz. Collective dynamics of'small-world'networks. *Nature*, 393(6684):440, 1998.

[36] Joseph Yose, Ralph Kenna, Pádraig Mac Carron, Thierry Platini, and Justin Tonra. A networks-science investigation into the epic poems of ossian. *Advances in Complex Systems*, 19(04n05):1650008, 2016.