2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

The one comparing narrative social network extraction techniques

Michelle Edwards
School of Mathematical Sciences, ACEMS
The University of Adelaide
SA 5005 Australia
michelle.edwards@adelaide.edu.au

Jonathan Tuke
School of Mathematical Sciences, ACEMS
The University of Adelaide
SA 5005 Australia
simon.tuke@adelaide.edu.au

Matthew Roughan
School of Mathematical Sciences, ACEMS
The University of Adelaide
SA 5005 Australia
matthew.roughan@adelaide.edu.au

Lewis Mitchell
School of Mathematical Sciences, ACEMS
The University of Adelaide
SA 5005 Australia
lewis.mitchell@adelaide.edu.au

***Abstract*—Analysing narratives through their social networks is an expanding field in quantitative literary studies. Manually extracting a social network from any narrative can be time consuming, so automatic extraction methods of varying complexity have been developed. However, the effect of different extraction methods on the resulting networks is unknown. Here we model and compare three extraction methods for social networks in narratives: manual extraction, co-occurrence automated extraction and automated extraction using machine learning. Although the manual extraction method produces more precise results in the network analysis, it is highly time consuming. The automatic extraction methods yield comparable results for density, centrality measures and edge weights. Our results provide evidence that automatically-extracted social networks are reliable for many analyses. We also describe which aspects of analysis are not reliable with such a social network. Our findings provide a framework to analyse narratives, which help us improve our understanding of how stories are written and evolve, and how people interact with each other.**

***Index Terms*—social networks, narratives, television**

I. Introduction

Quantitative narrative analysis has become increasingly popular in recent years with the availability of literary works, film and television scripts, online. Reasons to analyse films [1]–[3], television shows [4]–[6] or novels [7]–[12] include:

- to gain a deeper understanding of a particular narrative, narratives of a certain type, or narratives in general, or
- to help determine what would improve narratives in the future.

Narrative analysis is popular within blogs, where fans visualise data from television shows such as Game of Thrones [13], Seinfeld [14], The Simpsons [15], Grey's Anatomy [16] and Friends [17], [18]. Similarly, films [19], [20] and plays [21], [22] have been analysed quantitatively.

Fortuin *et al*. [23] used narrative analysis to inspire scriptwriters suffering from writer's block, while Gorinski *et al.* [24] analysed film scripts to find a logical chain of important events, allowing them to summarise film scripts automatically. We can also use narrative analysis to predict what will happen next [8]. Event prediction in narratives also suggests potential methods for predicting real-world events from news [25], [26].

A popular way of analysing narratives is through the social networks they describe. A social network for a narrative is comprised of characters (as nodes) and their interactions or relationships (as edges). As narratives are stories about characters' interactions [7], it makes sense to analyse the narrative by analysing how the characters interact. Understanding and comparing narrative social networks could lead to insights into which structures make a narrative successful.

One of the most problematic aspects of narrative social network analysis is constructing the network from an unstructured text source such as a script or novel. Extracting an interaction network from novels is challenging because the text does not always state who is speaking. Most attempts to match quoted speech in novels to the character speaking involve Natural Language Processing (NLP) and/or machine learning techniques [9], [27]–[31]. A disadvantage of these techniques is that there is either significant manual work in identifying aliases of characters, or that the accuracy of character identification ranges from $< 50\%$ to $\approx 90\%$ [27]. The more manual work put in at the NLP stage, the more accurate the identification tends to be. Alternatively, researchers can manually identify the speakers in novels [32], but this takes substantially longer and is not practical for analysing large corpora.

Extracting social networks from film or television scripts is almost as difficult. The most accurate, but time-consuming, approach is to manually record interactions between characters [5]. A more scalable approach is to automatically create a social network from the script of the film or television show. Scripts necessarily label speakers, but not who each character is speaking to. There are examples of using NLP and machine learning techniques [33]–[35], but again there is a trade-off with the accuracy of identifications.

An alternative automatic method is to extract a co-

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

occurrence network [6], [36]–[38], which infers interactions between characters from the number of times they appear in a scene together. We can create co-occurrence networks for novels as well, for example by counting the number of times characters are mentioned within a number of words of each other [8]. Using a co-occurrence network presumes that relationship strength can be measured by the number of times characters share a scene, as opposed to the number of times characters directly interact. While this assumption is intuitive, to the best of our knowledge, there is no research into the effect of this assumption on the resulting network properties in narrative analysis.

In this paper we compare three social network extraction techniques in the context of TV scripts:

- manually-extracted networks (as in Bazzan [5]),
- networks extracted using NLP (as in Deleris *et al.* [35]), and
- co-occurence networks extracted using scripts.

To compare these techniques we create a model to simulate interactions in a narrative. Using the simulated interactions, we create and compare observation networks based on the three extraction techniques. This *in silico* model allows us to compare techniques with complete knowledge of the ground truth. Modelling the narrative also allows us to control and measure parameters such as the error rate for the NLP method or the number of scenes for the co-occurrence method. Finally, the model allows our methods to be applied to a range of narratives, not just the case study we give here.

We use standard network metrics (see subsection III-F) to compare the three different network extraction techniques, applied to the characters in the television series *Friends*. *Friends* is an American situation comedy (sitcom) with ten seasons aired from 1994 to 2004. We choose *Friends* as a case study for our model because the series is well-known, long-running and a popular subject amongst researchers [5], [6], [34], [35], [39].

Some key findings of this work are:

- Co-occurrence networks have higher edge densities than the manually extracted networks, but the densities are highly correlated between techniques (the Pearson's correlation coefficient is 0.96).
- Centrality measures (degree, betweenness, eigenvector and closeness) are highly correlated in the manually extracted networks and co-occurrence and NLP networks, but clustering is not reliable in the automated networks.
- Edge weights in the automated networks correlate moderately with the edge weights in the manually extracted networks (the median Spearman's correlation coefficient is 0.77 for the co-occurrence networks and 0.80 for the NLP networks).
- The six core "friends" in *Friends* (Chandler, Joey, Monica, Phoebe, Rachel and Ross) interact less with each other less as the series progresses.

We conclude that automatically extracted networks – co-occurrence and NLP networks – give reliable analyses for most global, character, and relationship metrics, so we recommend extracting narrative social networks in one of these ways for time efficiency. If clustering is of high importance in an analysis, however, manually extracted networks are required.

II. Data

Although our findings are partially based on *in silico* experiments, we use real data to inform our models and to provide final verification.

We examine three datasets estimating social networks for the television series *Friends*. The social network describing character relationships are defined by nodes that represent characters in a chosen time frame (usually an episode or season), and edges connecting characters who interact. The precise definition of an interaction varies throughout the literature, but the assumption that characters who interact more have stronger relationships remains constant. Our goal is to model these relationships. Note that the strength of a relationship does not imply characters are good friends (despite the name of the series), as characters can have strong hostile interactions [7], [40].

The first dataset consists of manually extracted data by Bazzan [5], available at https://github.com/anabazzan/friends. It contains ordered lists of undirected interactions between pairs of characters for each episode. Bazzan manually annotated 16569 interactions from all 236 episodes, defining an interaction as two characters talking, touching or having eye contact. While there may be human interpretation errors in this dataset, this is the most reliable method of extracting the social network. Therefore, the manual extraction method provides a 'gold standard' for the social networks of the characters. We call the networks from this dataset the manually extracted networks, or **manual networks**. The edge weights in the **manual networks** correspond to the number of interactions between two characters in a given timeframe. Table I shows the number of episodes, interactions, scenes and characters in each season.

Season	Eps	Chars	Ints	Scenes	Ints/Ep	Scenes/Ep	Ints/Scene
1	24	126	2492	364	103.83	15.17	6.85
2	24	107	1815	314	75.62	13.08	5.78
3	25	98	1770	422	70.80	16.88	4.19
4	24	96	1598	438	66.58	18.25	3.65
5	24	92	1786	378	74.42	15.75	4.72
6	25	99	1491	387	59.64	15.48	3.85
7	24	81	1475	402	61.46	16.75	3.67
8	24	110	1220	356	50.83	14.83	3.43
9	24	101	1454	345	60.58	14.38	4.21
10	18	88	1468	238	81.56	13.22	6.17

TABLE I
SUMMARY OF DATA FROM MANUALLY COLLECTED DATASET [5]. FOR EACH SEASON WE HAVE THE NUMBER OF EPISODES, CHARACTERS (CHARS), TOTAL NUMBER OF INTERACTIONS (INTS), AND NUMBER OF SCENES (SCENES). WE ALSO CALCULATE THE NUMBER OF INTERACTIONS PER EPISODE (INTS/EPISODE), SCENES PER EPISODE (SCENES/EPISODE) AND AVERAGE INTERACTIONS PER SCENE FOR EACH SEASON (INTS/SCENE).

Table I shows there are 24 episodes in most seasons, but 25 episodes in Season 3 and Season 6 and only 18 episodes in

Season 10. Season 1 has notably more interactions than any other season, possibly due to the need to establish characters and relationships at the beginning of the series. We will discuss our findings on trends in network properties over all 10 seasons in subsection IV-D.

The second dataset contains co-occurrence networks, extracted using scripts available from a fan website [41]. We identified scene breaks and the characters that speak in each scene. For every scene, we assume every character interacts with every other character, so a scene is a "co-occurrence", and hence we call these networks **co-occurrence networks**. The edge weights correspond to the number of co-occurrences between characters. This dataset is available at https://figshare.com/projects/Friends_co-occurrence_networks/57479.

Note that only speaking characters are identified, even though there could be scenarios where characters appear or interact without speaking. Also, the scripts [41] were transcribed manually, so there are issues with inconsistency, typing mistakes and characters being referred to by different names (*e.g*. Ross, Ross Geller or Mr. Geller). To minimise these issues, we cleaned the scripts using manually defined regular expressions. The resulting **co-occurrence networks** dataset contains weighted edge lists for 227 episodes. The total number of episodes is less than for the **manual networks** because episodes with two parts (*e.g*. S1E16 - The One With Two Parts: Part One and S1E17 - The One With Two Parts: Part Two) are included as one episode in the **co-occurrence networks** dataset. The total number of interactions over all **co-occurrence network** episodes is 18574, which is surprisingly close to that of the **manual networks** dataset.

An NLP network dataset for *Friends* was not available, but Deleris *et al*. [35] provide information about how they extracted the social network, making use of Chen and Choi's data [34]. Chen and Choi use NLP techniques to identify which character is mentioned when another character says 'you', 'he', 'they', etc. They estimate their model correctly identifies a character 69.21% of the time. Deleris *et al*. use 'character mention' information to build a directed social network where the interactions are one of four kinds of signals:

- Direct Speech (*e.g*. A talks to B).
- Direct Reference (*e.g*. A says 'I like you' to B).
- Indirect Reference (*e.g*. A says 'I like B').
- Third-Party Reference (*e.g*. C says 'A likes B').

Each of these is an example of a directed interaction from A to B. However, for the purpose of modelling networks consistently between all three approaches, we assume all interactions are reciprocated. We call the undirected networks extracted using this approach the **NLP networks**.

III. Method

A. Overview

We compare the network extraction methods by simulating narrative social networks, 'extracting' observed networks using the three extraction methods and comparing these observed networks. We simulate social networks using a data-driven model. Simulation allows us to generalise the problem to any narrative that has a similar underlying social network and to generate large datasets for statistical analyses. The simulation and extraction process is outlined in Figure 1 and the following sections.

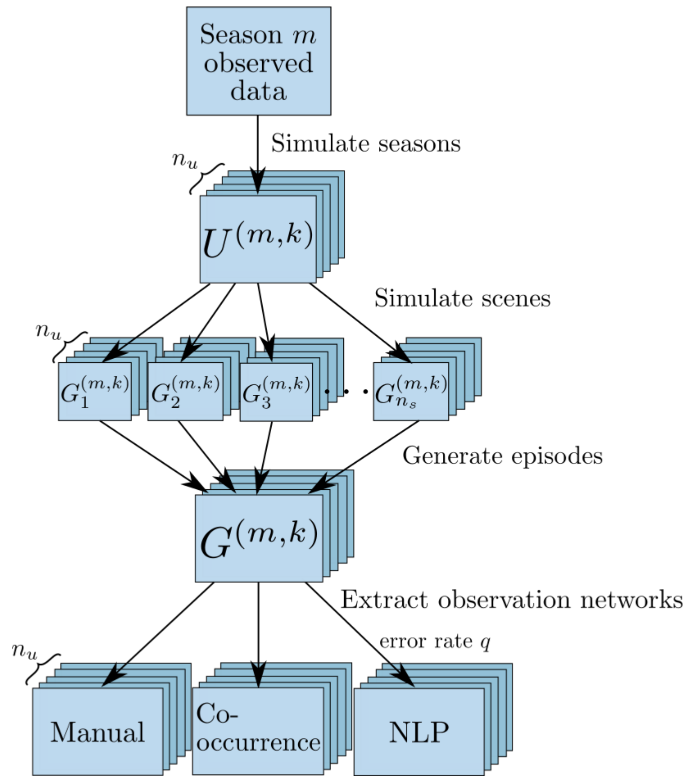

Fig. 1. Simulation process: We model each season m in the observed data to simulate underlying season networks $U^{(m,k)}$, where $k = 1, \ldots, n_u$ (subsection III-B). From each underlying season network we simulate scene networks $G_\ell^{(m,k)}$ for $\ell = 1, \ldots, n_s$ (subsection III-C). The scene networks combine to generate episode networks $G^{(m,k)}$ (subsection III-D). From each simulated episode network we extract three observation networks; a **manual network**, a **co-occurrence network** and an **NLP network** (subsection III-E).

We estimate parameters for our model using the **manual networks** data, then use the model to simulate n_u underlying season networks. For each season network, we use a random walk process to simulate n_s scenes. We then combine the scenes to form a simulated episode. From each simulated episode, we extract three observation networks resembling the **manual networks**, **co-occurrence networks** and **NLP networks**. We compare these simulated networks using the network metrics outlined in subsection III-F.

B. Simulate season from data

The first step described by Figure 1 is simulating underlying season networks from the observed data. To simulate networks we need to model the seasons in the **manual networks** dataset. We want, in addition to edges, to simulate edge weights, non-negative integers representing the number of character interactions. We notice there are significant differences between the way the core characters of *Friends* (Monica, Rachel, Phoebe, Ross, Chandler and Joey) interact with each other (average of 81 interactions per pair per season) and with other characters (average of 0.71 interactions per pair per season), and the way other characters interact with each other (average of 0.0093 interactions per pair per season). We therefore propose a two-class Poisson model for each season of the **manual networks**.

Let $V^{(m)} = \{1, \ldots, N^{(m)}\}$ be the set of characters in Season m and $w_{ij}^{(m)} \geq 0$ be the number of interactions between character i and character j in Season m of the **manual networks** dataset. We partition $V^{(m)}$ such that

$$V^{(m)} = V_{\text{core}} \cup V_{\text{non-core}}^{(m)},$$

where V_{core} contains the 6 core characters who are constant across all seasons, and $V_{\text{non-core}}^{(m)}$ contains the $\left(N^{(m)} - 6\right)$ non-core characters for Season m.

For the two-class Poisson model, assume each edge weight $w_{ij}^{(m)}$ in Season m of the **manual networks** dataset is a random observation of

$$W_{ij}^{(m)} \sim \text{Poi}\left(\lambda_{C_i C_j}^{(m)}\right),$$

where

$$C_i = \begin{cases} 1 & \text{if } i \in V_{\text{core}}, \\ 0 & \text{if } i \in V_{\text{non-core}}. \end{cases}$$

We estimate $\lambda_{C_i C_j}^{(m)}$ using maximum likelihood estimation with the **manual network** edge weights:

$$\hat{\lambda}_{C_i C_j}^{(m)} = \begin{cases} \dfrac{\sum_{i<j} w_{ij}^{(m)} C_i C_j}{\sum_{i<j} C_i C_j} & \text{if } C_i C_j = 1, \\ \dfrac{\sum_{i<j} w_{ij}^{(m)} (1-C_i)(1-C_j)}{\sum_{i<j} (1-C_i)(1-C_j)} & \text{if } C_{ij} = C_i + C_j, \\ \dfrac{\sum_{i<j} w_{ij}^{(m)} (C_i + C_j - 2C_i C_j)}{\sum_{i<j} (C_i + C_j - 2C_i C_j)} & \text{otherwise.} \end{cases}$$

For each season m we simulate n_u season networks

$$U^{(m,k)} = \left(V^{(m)}, E^{(m,k)}\right),$$

where $k = 1, \ldots, n_u$ and

$$E^{(m,k)} = \left\{W_{ij}^{(m,k)} \mid i, j \in V^{(m)}, i < j\right\}.$$

Note that each simulation contains all characters $V^{(m)}$ from Season m, but the edge weights are randomised. We generate random edge weights between each pair of nodes from the distribution

$$W_{ij}^{(m,k)} \sim \text{Poi}\left(\hat{\lambda}_{C_i C_j}^{(m)}\right).$$

This method allows for edges with zero-weights. We take zero-weights to mean there are no interactions between the characters, which is equivalent to having no edge between the characters.

C. Simulate scene from season

Given an underlying season network $U^{(m,k)}$, we wish to sample an episode network, every episode being a sequence of scenes. Following Fortuin *et al.* [23], we define a scene as a story part with a constant set of characters in a constant location. This approximation allows a consistent comparison between methods. Each scene also contains a set of interactions, so we can form a social network for every scene. Interactions within a scene are dependent. For example, if Joey talks to Monica, it is likely that Monica will then talk to Joey. We capture this in the model by proposing a simple random walk model for interactions in each scene. Note that this approach is able to capture higher-order sequences of interactions, dependent on the size of the dataset.

The random walk model randomly picks a starting character in $V^{(m)}$, with probability proportional to the eigenvector centrality of the character in $U^{(m,k)}$ (see subsection III-F). This character randomly interacts with another character with probability proportional to the edge weight between the characters. That character randomly interacts with another character, selected in the same way. The random walk process continues until we reach $n_{\text{int},\ell}$ interactions. We choose $n_{\text{int},\ell}$ based on the average number of interactions per scene in the data from Table I. The next scene starts with a new random starting character so that we have a fresh set of characters in each scene.

Each scene ℓ consists of a set of characters $C_\ell^{(m,k)}$ and interactions $L_\ell^{(m,k)}$. Let $n_s^{(m)}$ be the rounded average number of scenes per episode in Season m from our datasets. We define the network of a scene ℓ sampled from $U^{(m,k)}$ as

$$G_\ell^{(m,k)} = \left(C_\ell^{(m,k)}, L_\ell^{(m,k)}\right),$$

where $\ell = 1, \ldots, n_s^{(m)}$. As shown in Figure 1, we independently simulate n_s scenes with the random walk model from each simulated season network $U^{(m,k)}$, then combine the scenes to generate a random episode.

D. Generate episode from simulated scenes

We generate an episode by concatenating simulated scenes. An episode sampled from $U^{(m,k)}$ is

$$G^{(m,k)} = \left(G_1^{(m,k)}, G_2^{(m,k)}, \ldots, G_{n_s}^{(m,k)}\right).$$

The set of characters in $G^{(m,k)}$ are the union of the sets of characters in the scenes, *i.e.* $\bigcup_{\ell=1}^{n_s} C_\ell^{(m,k)}$. The edge weight between character i and j in $G^{(m,k)}$ is the sum of the interactions between characters i and j in the scene networks, which is zero if at least one of i or j was not in the scene.

E. Extract observation networks from simulated episodes

As in Figure 1, we extract three observed networks from each simulated episode $G^{(m,k)}$; a **manual network**, a **co-occurrence network** and an **NLP network**. We compare these simulated networks using metrics outlined in subsection III-F.

The **manual network** is built from the actual data so it is assumed to be 100% correct. Therefore the simulated **manual network** extracted from $G^{(m,k)}$ is $G^{(m,k)}$.

The **co-occurrence network** is obtained by creating a clique for the characters in each scene. We add clique networks so that edge weights correspond to the number of scenes two characters are in together, as they would be in the automated process.

The **NLP network** counts interactions similarly to the **manual network**, however it simulates NLP by including errors in the identification of characters. We model these errors by assuming:

1) One character (the speaking character) has been identified correctly, but the character being spoken to may be misidentified with probability q.
2) An incorrectly identified character is equally likely to be any character in the episode except the speaking character or correct character.

Chen and Choi [34] obtained a "purity score" of 69.21% in their analysis of *Friends*, which they describe as the effective accuracy of character identification and hence we set $q = 0.3$. The impact of q as it changes is a potential direction for future work. We call the process of incorrect character identification "rewiring".

In practice, the definition of an interaction (and hence edge weight) differs in **NLP networks** compared to **manual networks**. In the **manual networks** an interaction occurs between two characters who see, talk to or touch each other, whereas an interaction in the **NLP networks** occurs when two characters talk to, mention or refer to each other. We do not have this information in the **manual networks**, so we assume that characters seeing and touching each other is equivalent to characters mentioning and referring to each other.

F. Comparing observation networks

To measure how social network extraction method affects narrative analysis we compare the simulated **manual networks** with the simulated **co-occurrence** and **NLP networks**, using three types of network metrics;

1) *Global metrics*: size, total edge weight, edge density and clustering coefficient.
2) *Node/character metrics*: degree, betweenness centrality, eigenvector centrality, closeness centrality and local clustering coefficient.
3) *Edge/relationship metrics*: edge weights.

These metrics are common in narrative social network analysis, providing useful insight into social structure and important characters and relationships. The aim is not to compare the observation networks directly, but to investigate the effect the different observation types have on the narrative analysis. Consequently, we are more interested in understanding how metrics correlate rather than systematic differences in their value.

IV. Results

We simulate $n_u = 10000$ seasons using the two-class Poisson model on Season $m = 6$, as the number of scenes per episode in Season 6 is close to the mean number of scenes per episode over all ten seasons. Figure 2 shows the social network of interactions from Season 6. From each simulation, we sample interactions from one episode using random walks for each scene. Table I shows Season 6 of Friends has 25 episodes, 1491 interactions and 379 scenes. Therefore the average episode in season 6 has approximately 60 interactions and 20 scenes. We set $n_s = 15$ scenes and $n_{int,\ell} = 4$ for every scene ℓ.

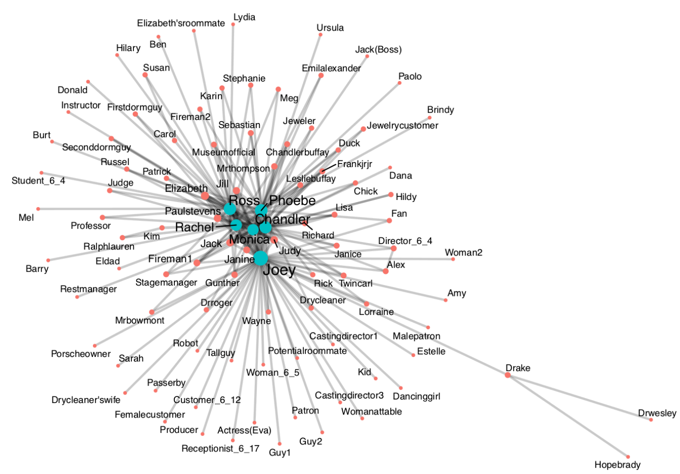

Fig. 2. Network of Season 6 from the manually extracted network dataset. The core characters have blue nodes and other characters have red nodes. The width of the edges represent the edge weight and the size of the nodes represent the node degree.

From each sample episode network we 'extract' the three observation networks using the methods described in subsection III-E and compare using the metrics described in subsection III-F. We find that there are differences in the value of the metrics across observation networks, but the errors are mostly systematic. We also checked that summaries of the extracted networks such as the edge weight distributions are also similar to their empirical counterparts. While the exact values of metrics can vary across the different observation networks, the important features in the narrative analysis (*i.e.* the rankings and trends of metrics) would not be greatly affected. The global metrics of the simulated **co-occurrence networks** and **NLP networks** correlate to those of the associated **manual networks**. The centrality metrics (degree, betweenness, eigenvector and closeness centrality) also have high correlation with the same character metrics across the simulated **manual** and random observation networks, but there is wide variance in the correlations of the local clustering coefficient of characters. The edge weights in the simulated **manual networks** also correlate reasonably highly with the edge weights in the simulated **co-occurrence** and **NLP networks**.

A. Global metric comparison

Figure 3 show boxplots of the normalised size, total edge weight, density and clustering for each network type. We normalise size and total edge by dividing by the maximum over the three network types.

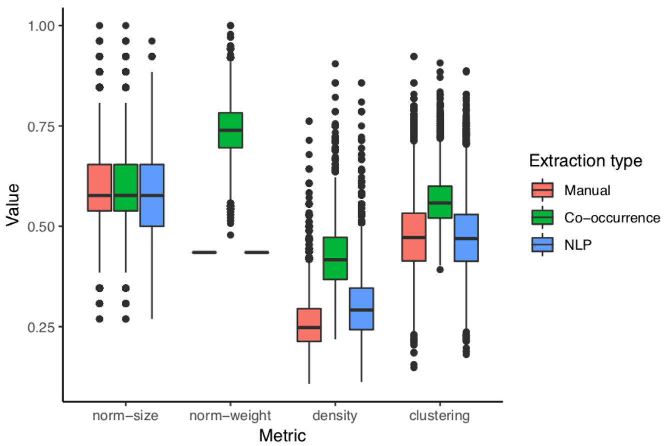

Fig. 3. Box-plots of the normalised size, normalised total edge weight, edge density and clustering coefficients of the **manual**, **co-occurrence** and **NLP networks** from the 10000 simulations. The size and total edge weights are normalised by dividing by the maximum.

The simulated **manual** and **co-occurrence networks** have the same number of characters (and hence size) in each simulation. However, it is possible in real data to see differences. We find that the discrepancies in size of the real datasets are almost always due to differences in what defines a character. For example, 'answering machine' is counted as a character in one **co-occurrence network**, but not in the corresponding **manual network**. The size of the simulated **NLP network** is always equal to or less than the size of the other simulated observation networks, as our model can only rewire to characters within the true episode. Characters are excluded if all the edges connected to that character are rewired away and no edges are rewired back to the character. This is more likely to happen to characters that are connected to few edges in the first place, and so the effect of the rewiring on the analysis is minimal.

The simulated **manual** and **NLP networks** have the same number of interactions (and hence total edge weight) in each simulation by construction. In practice there might be discrepancies in total edge weights due to different definitions of interactions as discussed in section II and subsection III-E. The simulated **co-occurrence networks** have between 10% and 100% more interactions on average. We see this in Figure 3 through an increase in the normalised weight and edge density.

Interestingly, it is rare for the simulated **NLP network** to have a lower edge density than the simulated **manual network** even though the edges are rewired with equal probability to any other character. This occurs because we only rewire one interaction, not the entire edge with its weight.

However, there is very high correlation between the **manual network** density and the other observation networks. The Pearson correlation coefficient between the simulated **manual** and **co-occurrence network** edge densities is 0.96, and between the simulated **manual** and **NLP network** edge densities is 0.95. While the co-occurrence networks are systematically denser than the NLP networks, comparing social networks using relative edge density is not greatly affected. Importantly, the different extraction methods don't distort trends.

Figure 3 shows that the simulated **co-occurrence networks** are more clustered than the simulated **manual networks**. This is not surprising as forming cliques for every scene creates clusters. Figure 4 shows a scatterplot of the relationships, showing that the increase in clustering from the simulated **manual** to **co-occurrence network** is smaller for more highly clustered networks. This occurs because if the **manual network** is already highly clustered, forming cliques in every scene will add fewer interactions between characters. Unclustered networks, however, will appear clustered using the co-occurrence method, so analysis of clustering is not reliable in **co-occurrence networks**. This is the largest non-systematic distortion we see across the different techniques.

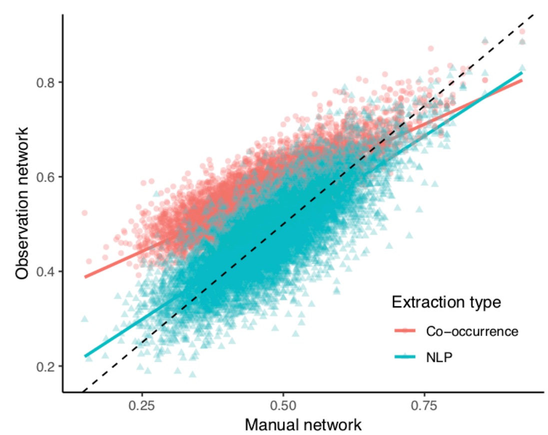

Fig. 4. Clustering coefficient of the simulated **manual network** containing all interactions compared to simulated **co-occurrence** and **NLP networks**. The dashed line shows $y = x$. The clustering coefficients of the **co-occurrence networks** are larger than the **manual networks** because we create cliques in every scene. The clustering of the simulated **NLP networks** is similar to the clustering of the **manual networks** but there is variation.

The clustering coefficients of the simulated **NLP networks** are similar to the **manual networks** (Pearson's correlation coefficient of 0.80), but there is some variation due to rewiring interactions. Therefore, when analysing clustering in the networks, **NLP networks** are reliable in general.

B. Character metrics

Node/character metrics are used to investigate the role of each character in the narrative. For example, Agarwal *et al.* [32] determined that Alice was the main storyteller in *Alice in Wonderland*, whereas Mouse's main role was to introduce other characters to Alice. Similarly Bazzan [5] showed that in *Friends*, while Joey connects many characters, Monica interacts the most with the five other main characters. We use degree, betweenness, eigenvector, and closeness centrality, along with local clustering coefficient to assess relative character importance. As we care more about comparisons between characters *e.g.* who is the most central, we examine correlations between character metrics for different networks, rather than the actual values. We use Spearman's correlation coefficient to investigate character rankings. Figure 5 shows box-plots of these correlations for these metrics.

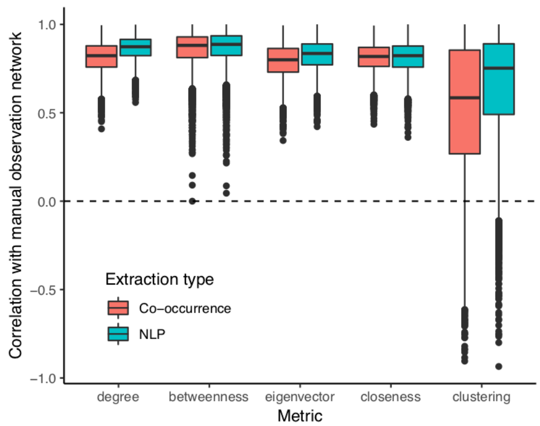

Fig. 5. Box-plots of the character metric correlations between the simulated **manual network** and simulated **co-occurrence** and **NLP networks** from the 10000 simulations. The degree, eigenvector and closeness centrality correlations are high with little variance. The betweenness centrality correlations are reasonably high but with more outliers. The clustering correlations have a large range and are extremely variable.

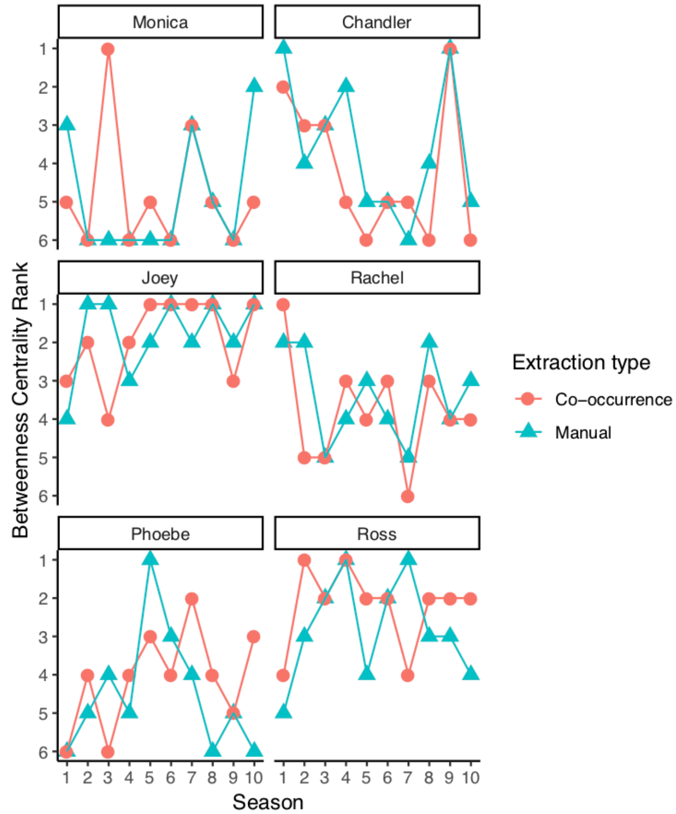

Fig. 6. The betweenness centrality ranks of all core characters over the 10 seasons of *Friends* for the **manual** and **co-occurrence network** datasets. Joey's betweenness centrality rank is very similar in every season, with a maximum difference of 2 in Season 10.

The network observation type affects each centrality similarly. Correlations between centrality rankings of characters in both simulated **co-occurrence** and **NLP networks** and simulated **manual networks** are high (approximately 0.8) on average, with little variance, meaning observation type does not have a strong effect on character ranking centrality.

We see a similar pattern in the data. Figure 6 shows betweenness centrality rankings for each character in the **manual** and **co-occurrence** season networks. Joey has the highest or second highest betweenness ranking in every season except Seasons 1 and 4 (and Season 10 in the **co-occurrence network**). While the exact value of the centrality may change in the different datasets, the rankings are similar, so analysis of character importance would be independent of network type.

The local clustering coefficient however is highly variable in the simulated **co-occurrence** and **NLP networks**. The correlations between the simulated **manual networks** and observation networks are moderate and positive on average, but range from -0.93 to 1. The large range of correlations show that we should not trust automatically extracted networks when looking at local clustering coefficients.

C. Relationship metrics

We use edge weights to investigate the importance of relationships between characters. We correlate three sets of edge weights to asses the accuracy of different types of relationship analyses:

1) All weights - including zero weights where there is no edge,
2) Non-zero weights - between characters that interact at least once in at least one of the networks, and
3) Core weights - between core characters, as these are usually the relationships we are most interested in.

Figure 7 shows the correlations between edge weights for the simulated **co-occurrence** and **manual networks** and simulated **NLP** and **manual networks**. Again, we use Spearman's correlation coefficient because we are interested in rank orderings rather than actual values.

The correlation between all edge weights in the simulated **manual** and **co-occurrence** networks are high with little variance. There is more spread in correlation between edge weights in the simulated **NLP** and **manual networks**. Correlations decrease when we exclude zero-weight edges in both.

But the non-zero edge weight correlations are still high for the simulated **co-occurrence networks**. This indicates that while we frequently get the correct set of interactions, the weights of those that interact are less accurate.

If we only compare the edges between the six core characters we find the correlations for both random observation networks vary greatly. Our results show the edge weight rankings in simulated **co-occurrence networks** are more reliable than in simulated **NLP networks** if we are interested in all characters. However, if we only analyse the relationships

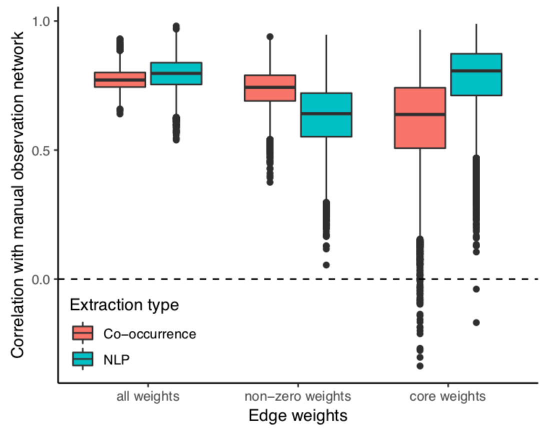

Fig. 7. Box-plots of the edge weight correlations between the simulated **manual network** and simulated **co-occurrence** and **NLP networks** from the 10000 simulations. The correlation is higher for both random networks when we include all edge weights. In both cases, the simulated **co-occurrence networks** are more reliable than the simulated **NLP networks**. If we only include edge weights between core characters there is a lot of variability in the correlations between the networks for both observation types.

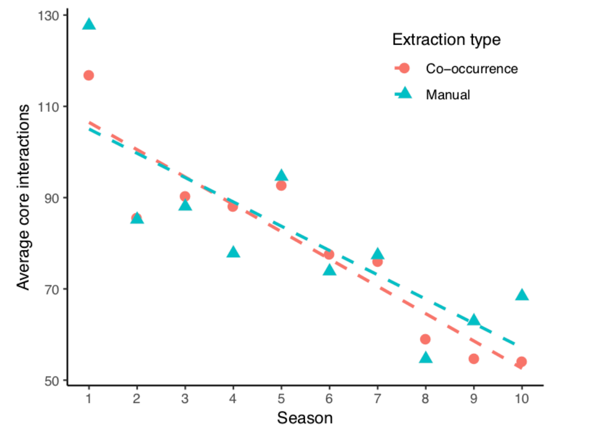

Fig. 8. Scatterplot of the average number of interactions between pairs of core characters in each season of *Friends* from the co-occurrence and manual datasets. The dashed lines represent the line of best fit for the co-occurrence and manual network average core interactions in red and blue respectively. We see that although the individual datapoints vary, the trend is preserved.

between core characters, the **NLP networks** are more reliable. This is because the core characters are frequently in scenes together but do not necessarily interact. This makes inferring the relative strengths of each relationship difficult when we only observe who is in the scene (*i.e*. from the **co-occurrence network**), but **NLP networks** misdirect each interaction with the same probability, so edge weights between core characters are equally likely to be changed.

D. Social networks in Friends

We now apply these network construction methods to real data. At http://friends-network.shinyapps.io/ingenuity_app/ we compare metrics from the real datasets to analyse the social networks in *Friends*. Here we focus on one finding, namely that the core *Friends* get less "friendly" over the ten seasons.

Figure 8 shows a scatterplot of the average number of interactions between pairs of core characters in each season of *Friends* as inferred from the co-occurrence dataset and the manual dataset. As the series develops, the core characters interact with each other less, *i.e*. the *Friends* get less friendly. This is consistent between both datasets, with only slight variations in the slope and variance of the line of best fit.

A possible explanation is that at the beginning of the series, relationships between core characters are established, meaning more core-core interaction. As the series develops, interactions between core characters become repetitive, so these characters interact with each other less, and extras become more important. An interesting direction for future research is the extent to which similar trends exist in other narratives.

V. Conclusion

The high correlation between the manually extracted network metrics and the automatically extracted networks suggest that for most narrative analyses we can extract the social network automatically and achieve similar results to the more time-consuming manual approach. However, this introduces some errors. For most metrics these have minimal effects on the *comparison* of global metrics over time, and character importance. A small set of metrics related to clustering are not reliable in automatically extracted networks.

We only investigated the effect of extraction method on one television show $Friends$. With these comparison methods, one could explore whether similar metrics appear in other television shows, films, and novels.

The core *Friends* are intrinsic to that series, but extending to other narratives means we have to identify core characters in those narratives. A Stochastic Block Model [42] could be used here to automatically identify the core group of characters. It is interesting to consider how the extraction approach might bias this identification, as some approaches to find core characters might be perturbed by distortion in clustering.

Acknowledgment

We thank A. Bazzan for sharing her data with us.

References

[1] S. Chaturvedi, S. Srivastava, and D. Roth, "Where Have I Heard This Story Before? Identifying Narrative Similarity in Movie Remakes," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ser. Short Papers, vol. 2, 2018, pp. 673–678.

[2] 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," *EPJ Data Science*, vol. 5, no. 1, p. 31, 2016.

[3] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Hierarchical neural network generative models for movie dialogues," *CoRR, abs/1507.04808*, 2015.

[4] X. Bost, V. Labatut, S. Gueye, and G. Linarès, "Narrative Smoothing: Dynamic Conversational Network for the Analysis of TV Series Plots," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 1111–1118.

[5] A. L. Bazzan, "I will be there for you: six friends in a clique," *arXiv preprint arXiv:1804.04408*, 2018.

[6] C.-J. Nan, K.-M. Kim, and B.-T. Zhang, "Social Network Analysis of TV Drama Characters via Deep Concept Hierarchies," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2015.

[7] S. Min and J. Park, "Mapping Out Narrative Structures and Dynamics Using Networks and Textual Information," 04 2016. [Online]. Available: https://arxiv.org/abs/1604.03029
[8] A. Beveridge and J. Shan, "Network of Thrones," *Math Horizons*, vol. 23, no. 4, pp. 18–22, 2016.
[9] M. C. Waumans, T. Nicodème, and H. Bersini, "Topology Analysis of Social Networks Extracted from Literature," *PloS one*, vol. 10, no. 6, p. e0126470, 2015.
[10] H. F. de Arruda, F. N. Silva, and V. Queiroz Marinho, "Representation of texts as complex networks: a mesoscopic approach," Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil. São Carlos Institute of Physics, University of São Paulo, São Carlos, SP, Brazil, Tech. Rep., 2017.
[11] P. Mac Carron and R. Kenna, "Viking sagas: Six degrees of Icelandic separation Social networks from the Viking era," *Significance*, vol. 10, no. 6, pp. 12–17, 2013.
[12] S. D. Prado, S. R. Dahmen, A. L. Bazzan, P. M. Carron, and R. Kenna, "Temporal Network Analysis of Literary Texts," *Advances in Complex Systems*, no. 03, p. 1650005, May.
[13] S. Glander. (2017, May) Network analysis of Game of Thrones family ties. Accessed: 1/3/18. [Online]. Available: https://shiring.github.io/networks/2017/05/15/got_final
[14] S. Stoltzman. (2016, December) Seinfeld Characters – A Post About Nothing. Accessed: 1/6/17. [Online]. Available: https://www.stoltzmaniac.com/seinfeld-characters-a-post-about-nothing/
[15] T. Schneider. (2016, September) The Simpsons by the Data. Accessed: 17/5/2017. [Online]. Available: http://toddwschneider.com/posts/the-simpsons-by-the-data/
[16] B. Lind. (2012, September) Lessons on exponential random graph modeling from Grey's Anatomy hook-ups. Accessed: 1/6/17. [Online]. Available: http://badhessian.org/2012/09/lessons-on-exponential-random-graph-modeling-from-greys-anatomy-hook-ups/
[17] G. Simchoni. (2017, June) The One With Friends. Accessed: 7/6/17. [Online]. Available: http://giorasimchoni.com/2017/06/04/2017-06-04-the-one-with-friends/
[18] Dmathlete. (2015, March) Friends and Hypergraphs: The One With All The Networks. Accessed: 7/6/17. [Online]. Available: http://mildlyscientific.schochastics.net/2015/03/03/friends-and-hypergraphs-one-with-a/
[19] E. Gabasova. (2015, December) The Star Wars social network. Accessed: 17/5/2017. [Online]. Available: http://evelinag.com/blog/2015/12-15-star-wars-social-network
[20] H. Anderson and M. Daniels. (2016, April) Film Dialogue from 2,000 screenplays, Broken Down by Gender and Age. Accessed: 1/6/17. [Online]. Available: https://pudding.cool/2017/03/film-dialogue/index.html
[21] M. Grandjean. (2015, December) Network visualization: mapping Shakespeare's tragedies. Accessed: 1/6/17. [Online]. Available: http://www.martingrandjean.ch/network-visualization-shakespeare/
[22] S. Wu. (2017, March) An Interactive Visualization of An Interactive Visualisation of Every Line in Hamilton. Accessed: 1/6/17. [Online]. Available: https://pudding.cool/2017/03/hamilton/index.html
[23] V. Fortuin, R. M. Weber, S. Schriber, D. Wotruba, and M. Gross, "InspireMe: Learning Sequence Models for Stories," in *AAAI Conference on Artificial Intelligence*.
[24] P. Gorinski and M. Lapata, "Movie Script Summarization as Graph-based Scene Extraction," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1066–1076.
[25] M. Granroth-Wilding and S. Clark, "What Happens Next? Event Prediction Using a Compositional Neural Network Model," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 2727–2733.
[26] Z. Li, X. Ding, and T. Liu, "Constructing Narrative Event Evolutionary Graph for Script Event Prediction," *ArXiv e-prints*, 2018.
[27] H. He, D. Barbosa, and G. Kondrak, "Identification of Speakers in Novels," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 1312–1320.
[28] D. K. Elson, N. Dames, and K. R. McKeown, "Extracting Social Networks from Literary Fiction," in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 138–147.
[29] A. Agarwal, A. Kotalwar, and O. Rambow, "Automatic Extraction of Social Networks from Literary Text: A Case Study on Alice in Wonderland," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 1202–1208.
[30] M. Iyyer, A. Guha, S. Chaturvedi, J. Boyd-Graber, and H. Daumé III, "Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1534–1544.
[31] A. Celikyilmaz, D. Hakkani-Tur, H. He, G. Kondrak, and D. Barbosa, "The Actor-Topic Model for Extracting Social Networks in Literary Narrative," in *NIPS Workshop: Machine Learning for Social Computing*, 2010.
[32] A. Agarwal, A. Corvalan, J. Jensen, and O. Rambow, "Social Network Analysis of Alice in Wonderland," in *Workshop on Computational Linguistics for Literature*, 2012, pp. 88–96.
[33] S.-Y. Chen, C.-C. Hsu, C.-C. Kuo, Ting-Hao, Huang, and L.-W. Ku, "EmotionLines: An Emotion Corpus of Multi-Party Conversations," *ArXiv e-prints*, Feb 2018.
[34] Y.-H. Chen and J. D. Choi, "Character Identification on Multiparty Conversation: Identifying Mentions of Characters in TV Shows," in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles: Association for Computational Linguistics, pp. 90–100.
[35] L. A. Deleris, F. Bonin, E. Daly, S. Deparis, Y. Hou, C. Jochim, Y. Lassoued, and K. Levacher, "Know Who Your Friends Are: Understanding Social Connections from Unstructured Text," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HTL 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Demonstrations*, pp. 76–80.
[36] M. Janosov. (2017, July) Network Science Predicts Who Dies Next in Game of Thrones. Accessed: 22/6/18. [Online]. Available: https://networkdatascience.ceu.edu/article/2017-07-08/network-science-predicts-who-dies-next-game-thrones
[37] P. Ercolessi, C. Sénac, and H. Bredin, "Toward plot de-interlacing in TV series using scenes clustering," in *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, June 2012, pp. 1–6.
[38] C.-Y. Weng, W.-T. Chu, and J.-L. Wu, "Rolenet: Movie Analysis from the Perspective of Social Networks," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 256–271, 2009.
[39] L. M. Marshall, ""I'll be there for you" if you are just like me: an analysis of hegemonic social structures in "Friends"," Ph.D. dissertation, Graduate College of Bowling Green State University, August 2007.
[40] P. Mac Carron and R. Kenna, "Universal properties of mythological networks," *EPL (Europhysics Letters)*, vol. 99, no. 2, p. 28002, 2012.
[41] (2004) Crazy For Friends. Accessed: 27/7/2017. [Online]. Available: http://www.livesinabox.com/friends/scripts.shtml
[42] C. Aicher, A. Z. Jacobs, and A. Clauset, "Adapting the Stochastic Block Model to Edge-Weighted Networks," *arXiv preprint arXiv:1305.5782*, 2013.